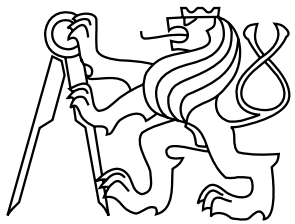
Programování grafických aplikací (BI-PGA), Přednáška č. 10

3D grafika - Blender: Datové struktury pro 3D grafiku

Jiří Chludil

Fakulta informačních technologií
České vysoké učení technické v Praze
https://courses.fit.cvut.cz/BI-PGA/



ZS 2020/2021

Motivace

V různých přednáškách se často zminňují následující pojmy

- konvexní obálka
- K-D Stromy
- zametací hyperrovina
- oktalový strom
- atd ...

Většina těchto algoritmů souvisí s efektivními algortimy viz. BI-AG1 dříve BI-EFA.

.

Tato přednáška vychází z podkladových materiálů, které vynikly pro BI-EFA (prof. Ing. Pavel Tvrdík, CSc., Ing. Jiří Chludil a Ing. Petr Matyáš).

Výpočetní geometrie

Definice

Cílem výpočetní geometrie je analýza a návrh **efektivních algoritmů** pro určování vlastností a vztahů geometrických objektů.
(bezrozměrných geometrických bodů, základních grafických primitiv, obecně jakýchkoli množin bodů ohraničených pomocí čar nebo ploch).

Řešené problémy:

- geometrické vyhledávání,
- reprezentace scény
- detekce kolize objektů
- konstrukce konvexní obálky,
- určení polohy objektů,

Výpočetní geometrie

Definice

Cílem výpočetní geometrie je analýza a návrh **efektivních algoritmů** pro určování vlastností a vztahů geometrických objektů.
(bezrozměrných geometrických bodů, základních grafických primitiv, obecně jakýchkoli množin bodů ohraničených pomocí čar nebo ploch).

Řešené problémy:

- geometrické vyhledávání,
- reprezentace scény
- detekce kolize objektů
- konstrukce konvexní obálky,
- určení polohy objektů,

Kvadrantový strom definice

- Autory Kvadrantových stromů (Quadtree) jsou Raphael Finkel a J.L. Bentley,

Definice (Kvadrantový strom)

Kvadrantový strom je 4-ární strom, který splňuje navíc následující podmínky:

1. každý vnitřní uzel reprezentuje jednu dekompozici 2D prostoru na čtyři navzájem disjunktní kvadranty resp. oblasti,
2. každý jeho vnitřní uzel má právě 4 potomky s identifikací SV, SZ, JV, JZ,
3. každý uzel obsahuje prostorová data definující dekompozici (liší se dle použití),
4. každý list má definovanou maximální kapacitu obsažených objektů, pokud je překročena, je list dekomponován.

Použití kvadrantového stromu

- tento strom je vhodný pro efektivní reprezentaci elementů (dané souřadnicemi x a y) v 2D prostoru např hmotné body objektů, což umožňuje:
  - snížení paměťových nároků při zobrazování 2D scény
  - urychlení 3D grafických algoritmů (sledování paprsku),
  - zpracování a analýzu obrazu,
  - efektivní uložení dat pro GIS (Geografické Informační Systémy),

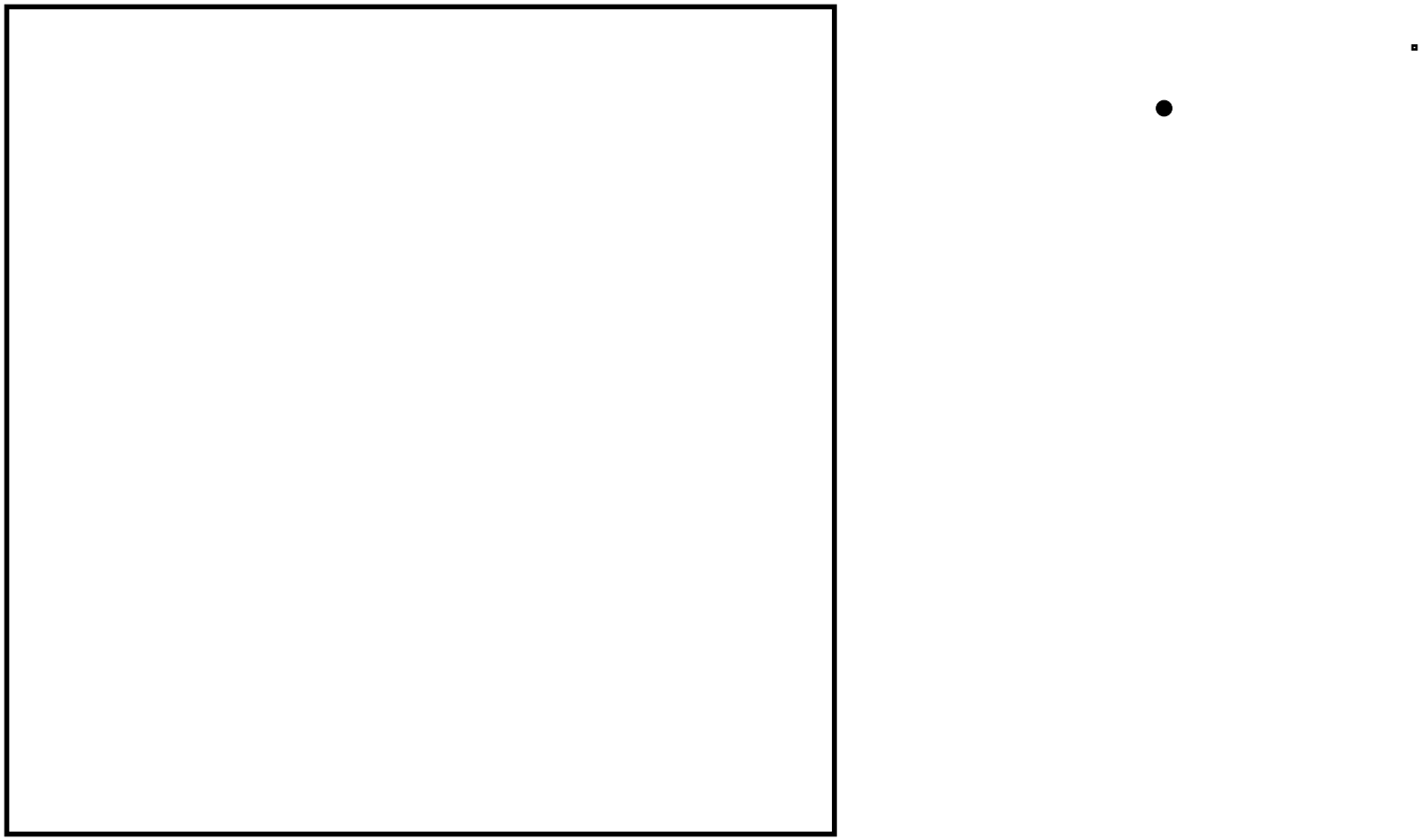
PR kvadrantový strom (Region Point quadtree)

Definice (PR kvadrantový strom)

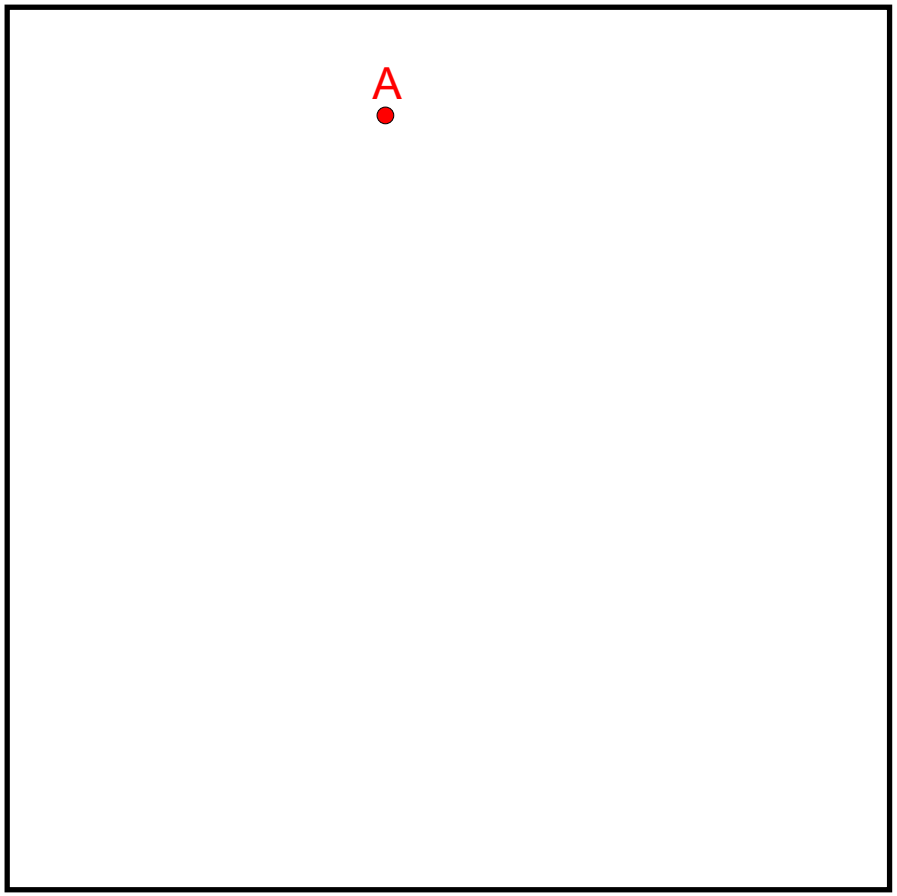
Je kvadrantový strom s těmito vlastnosti:

1. všechny dekomponované oblasti jsou stejně velké,
2. maximální kapacita pro obsažené elementy je 1,
3. uživatelská data jsou elementy a jsou uloženy jen v listech,
4. pro urychlení výpočtu obsahuje každý uzel souřadnice oblasti, kterou uzel reprezentuje.

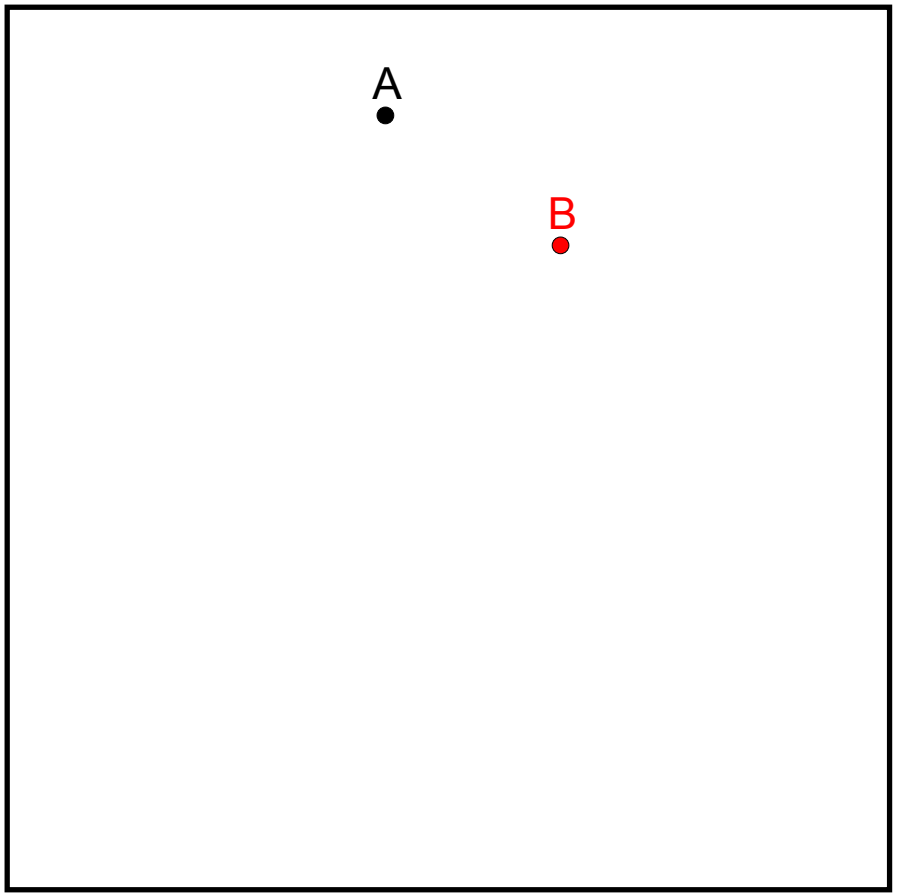
Příklad PR kvadrantového stromu



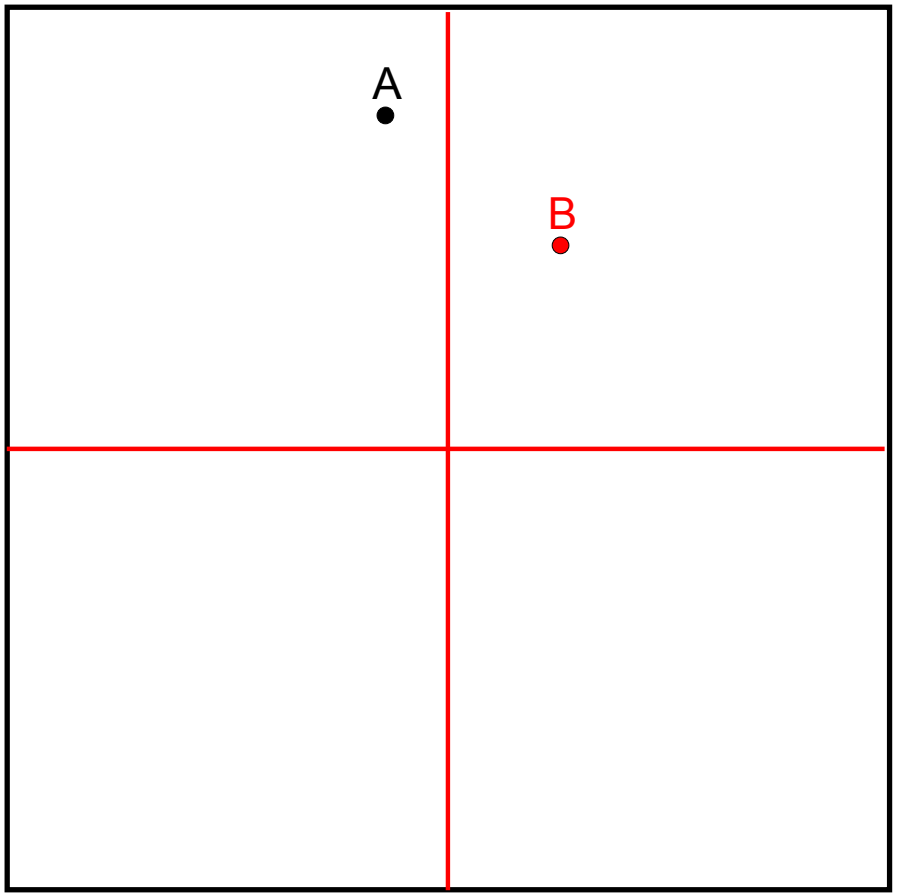
Příklad PR kvadrantového stromu

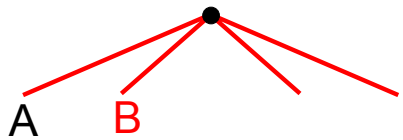


Příklad PR kvadrantového stromu

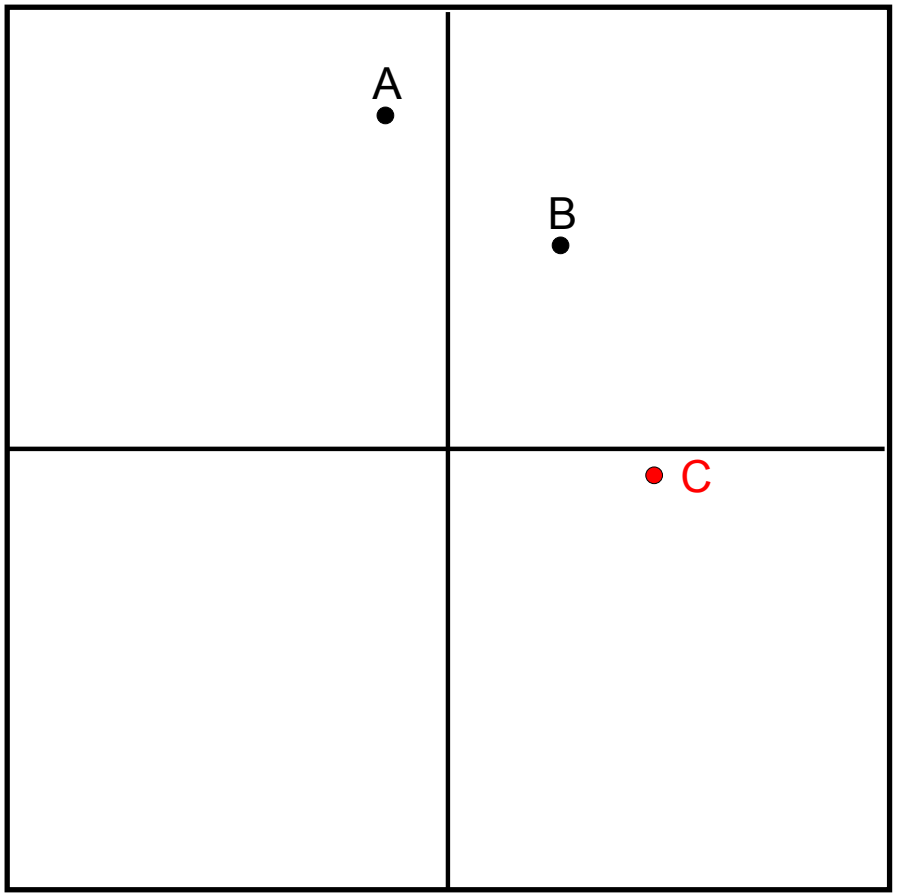


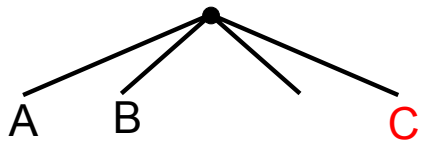
Příklad PR kvadrantového stromu



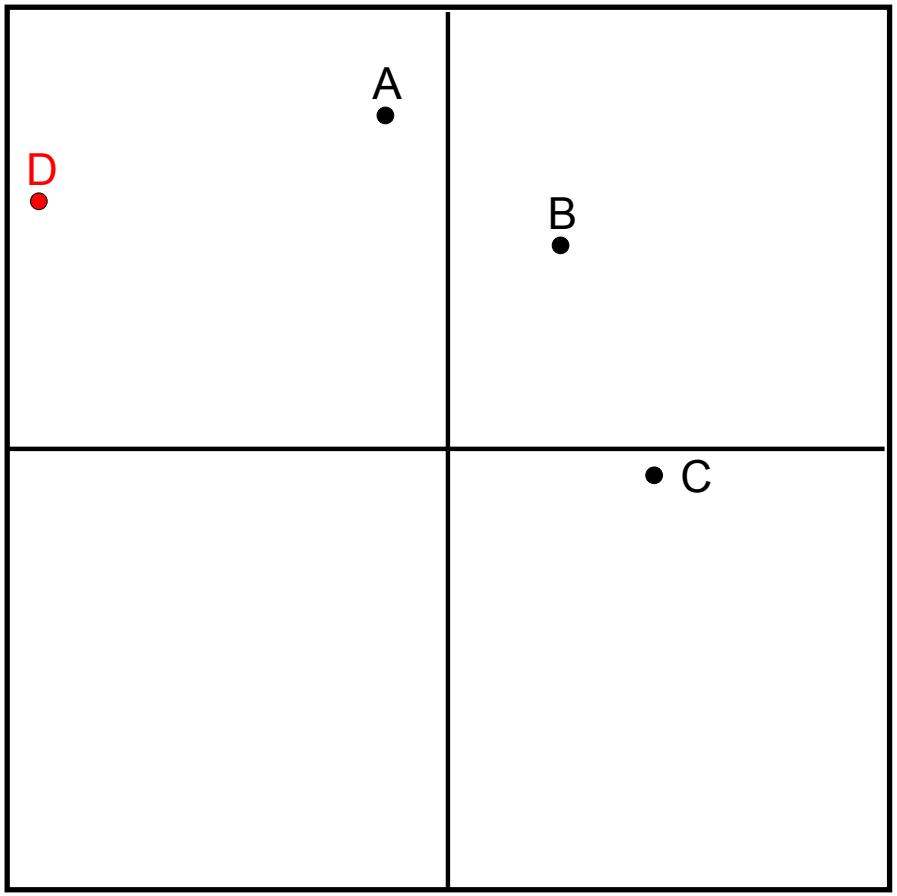


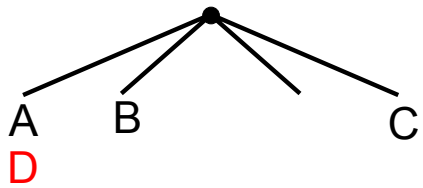
Příklad PR kvadrantového stromu



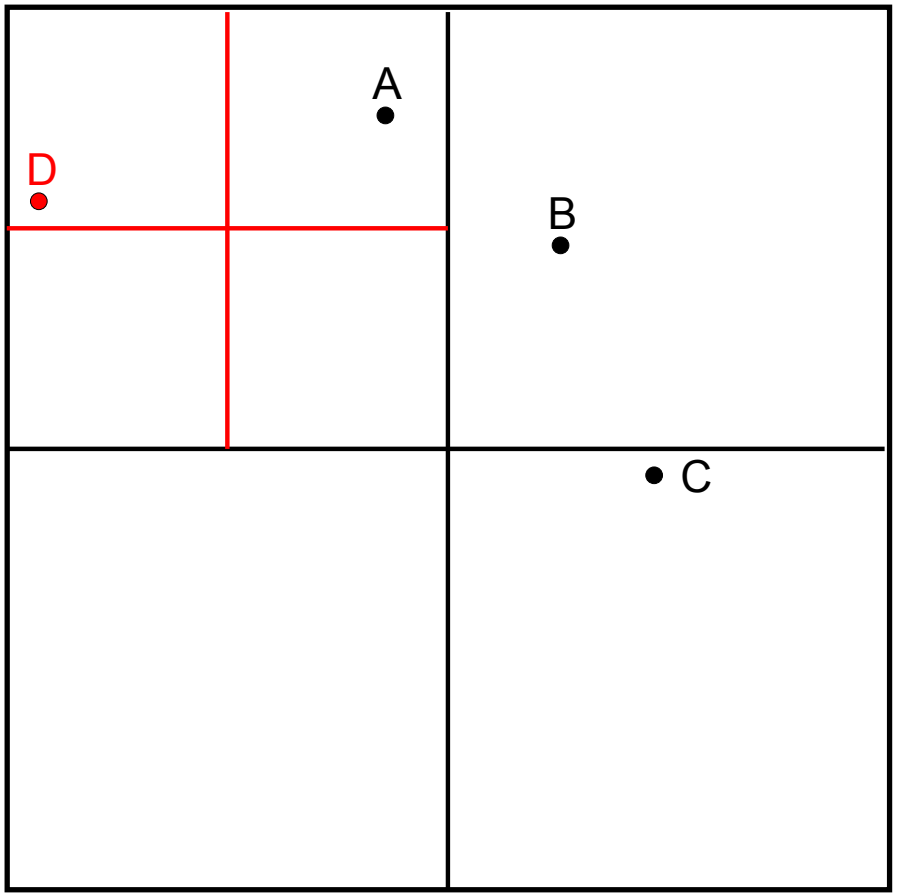


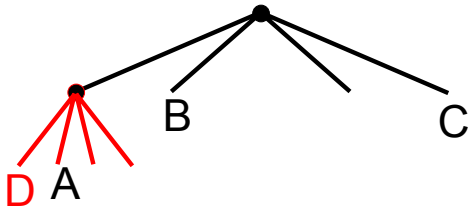
Příklad PR kvadrantového stromu



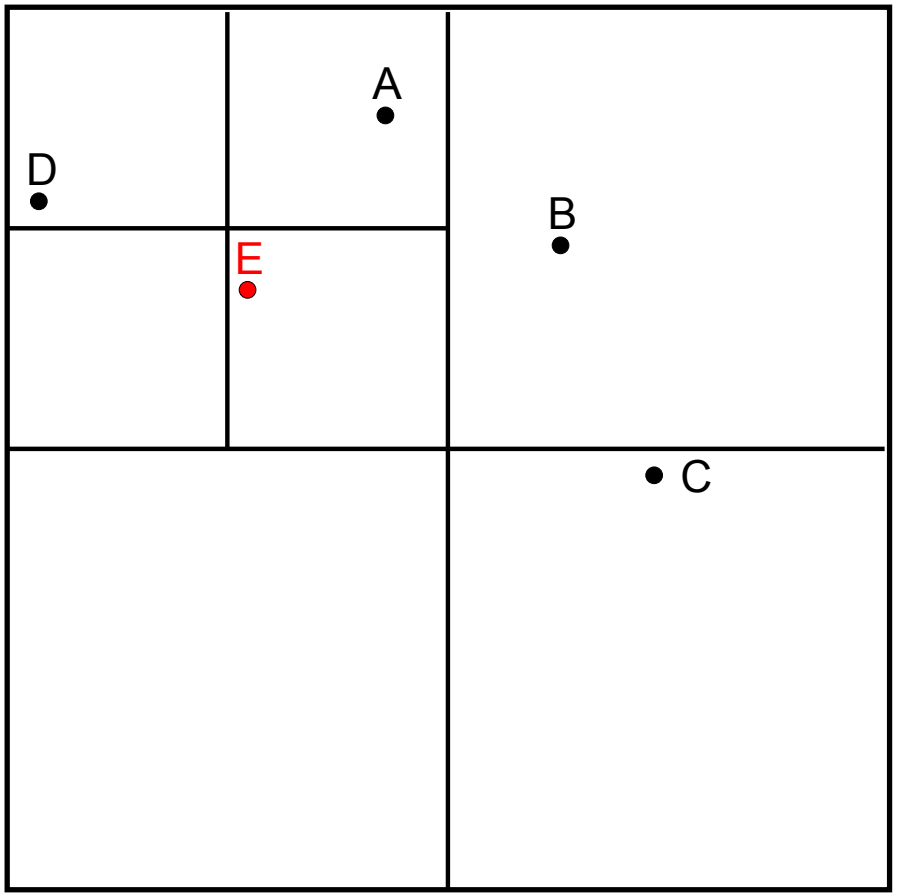


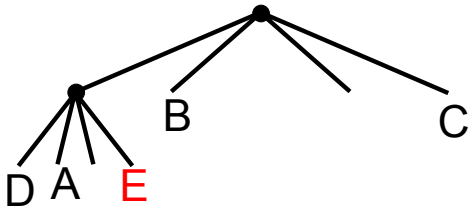
Příklad PR kvadrantového stromu



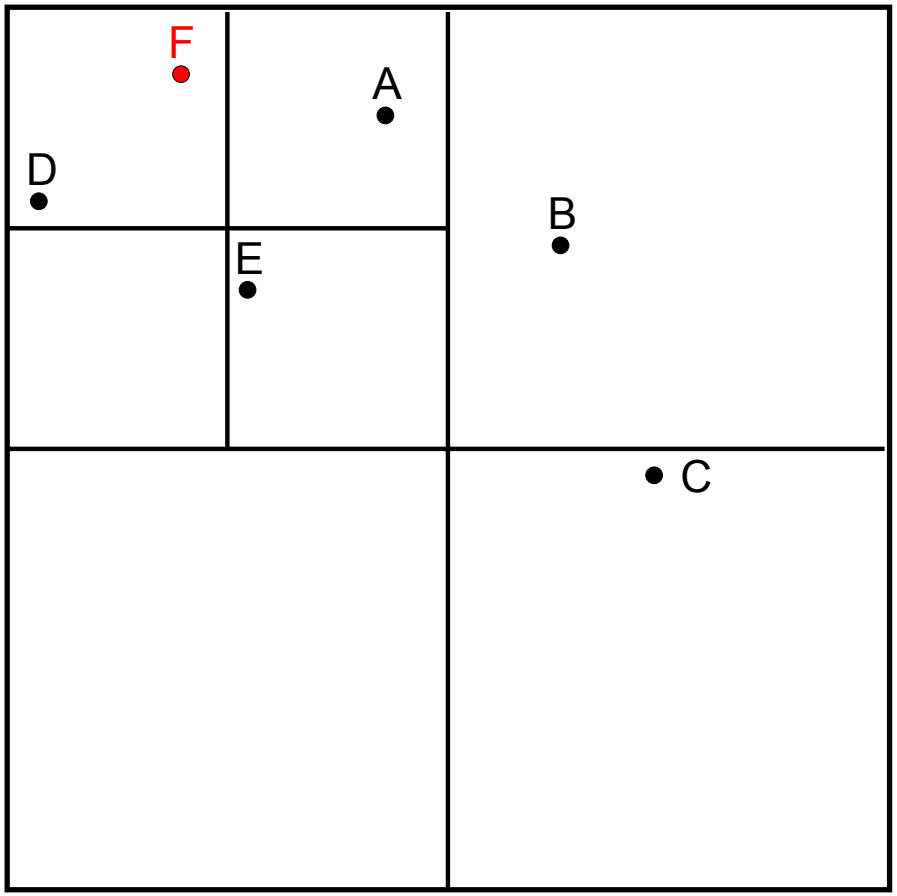


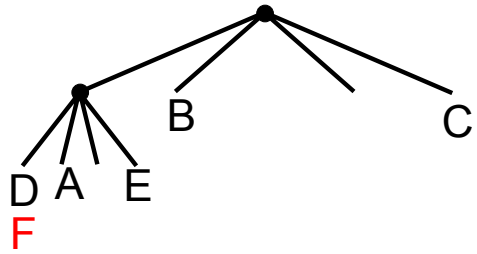
Příklad PR kvadrantového stromu



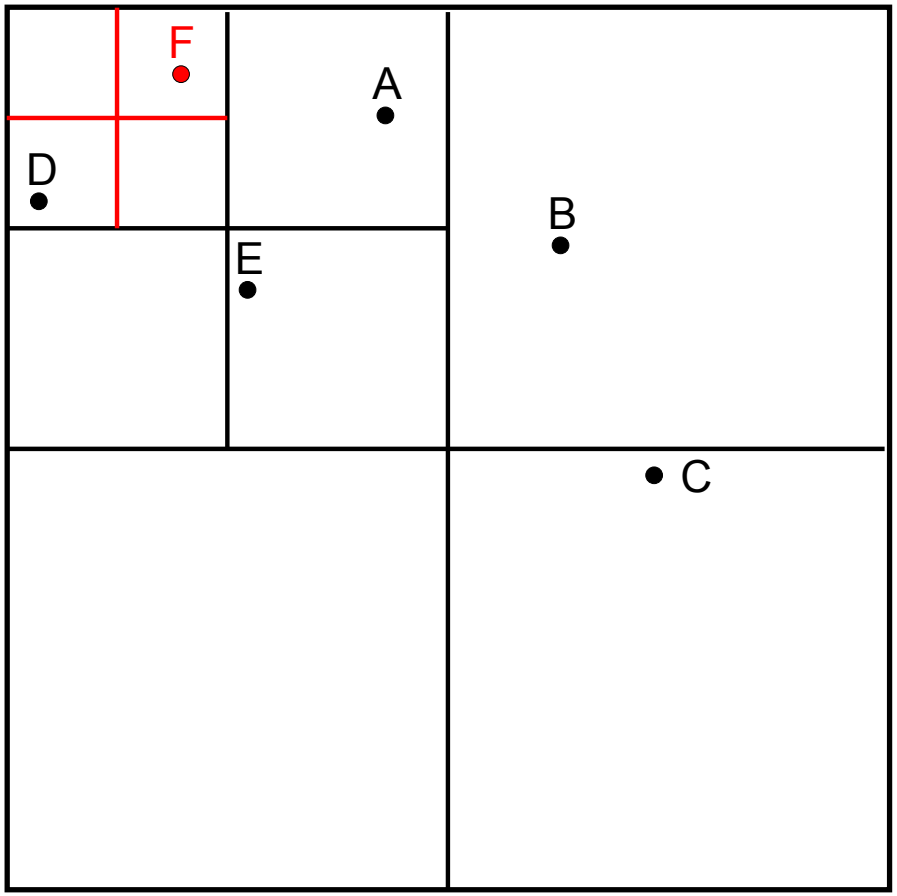


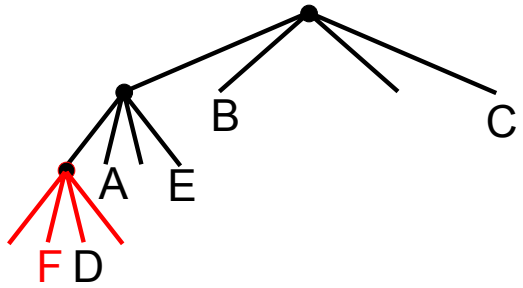
Příklad PR kvadrantového stromu



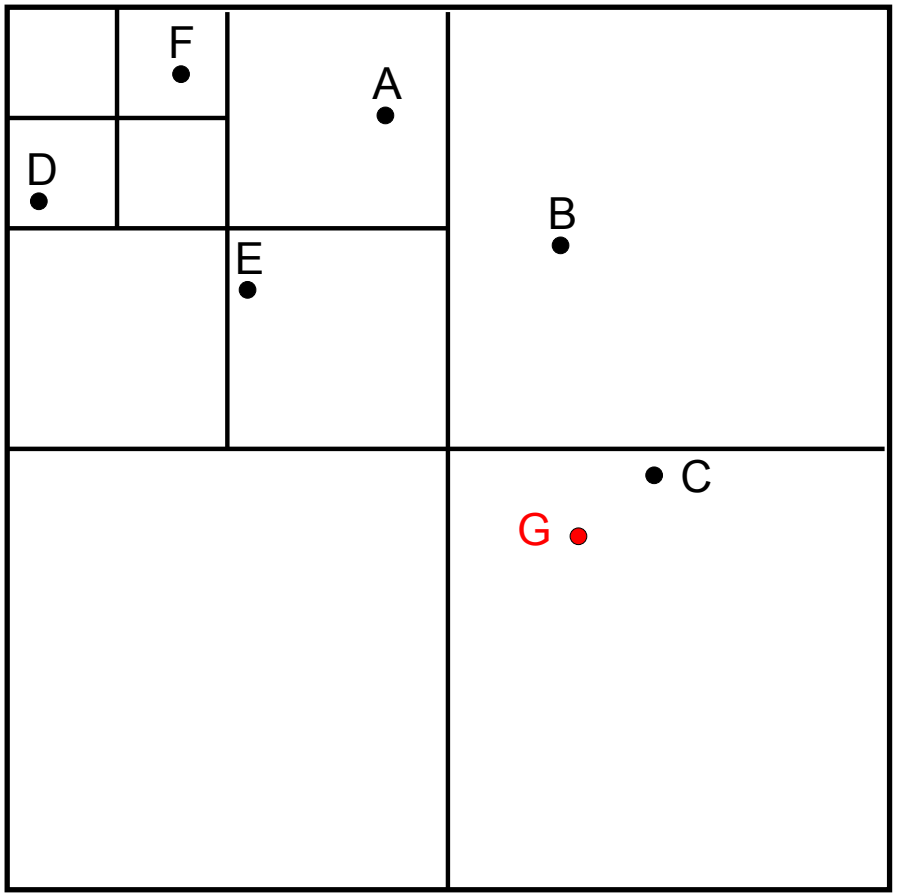


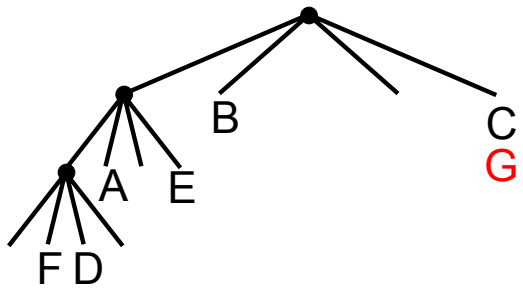
Příklad PR kvadrantového stromu



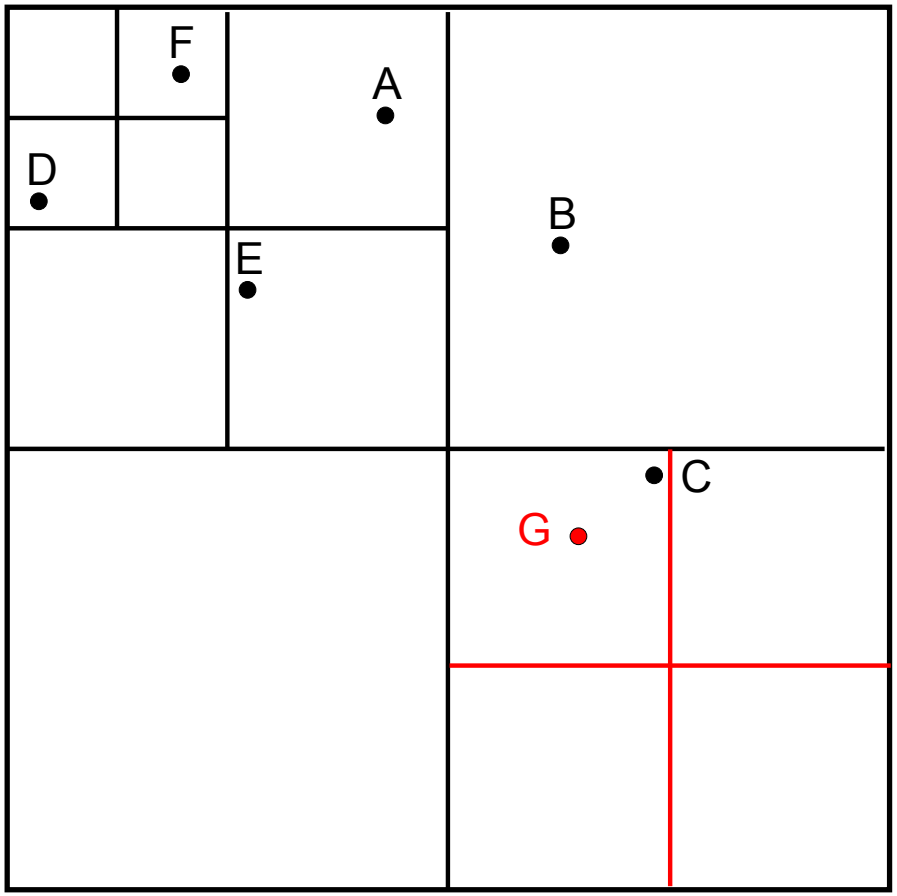


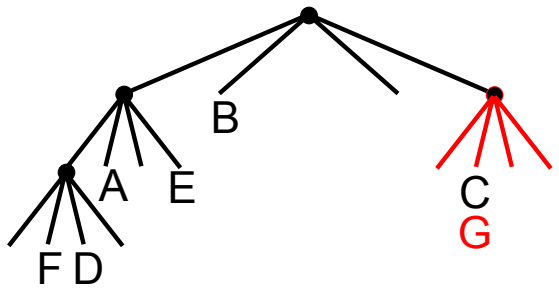
Příklad PR kvadrantového stromu



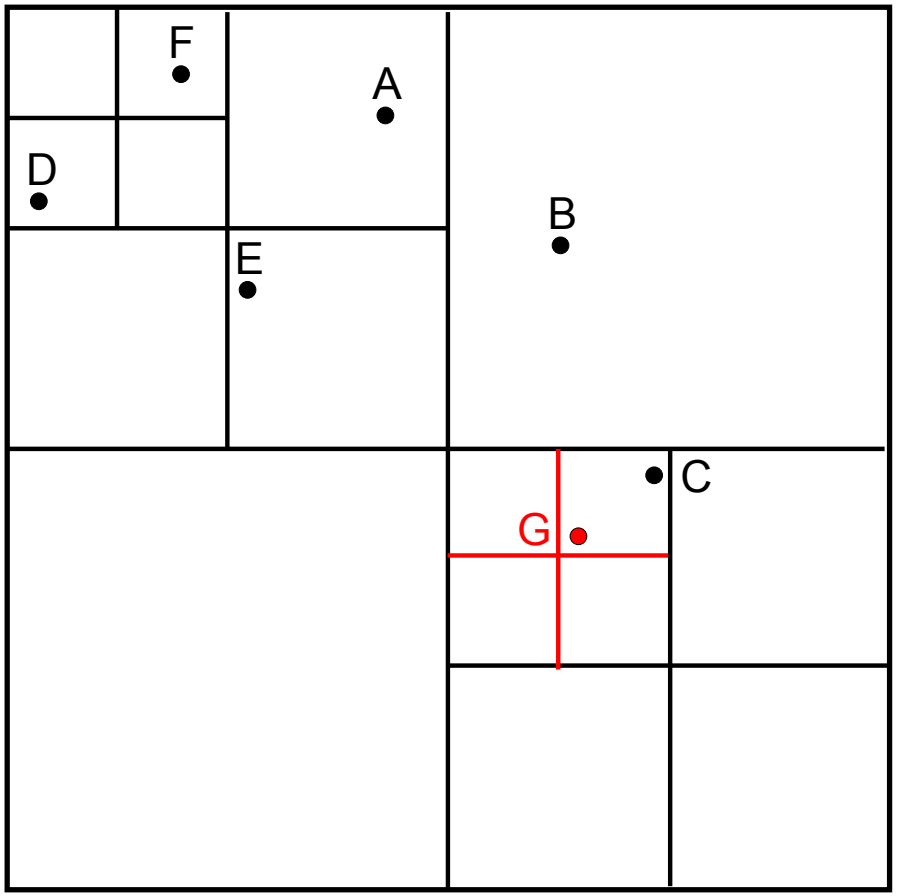


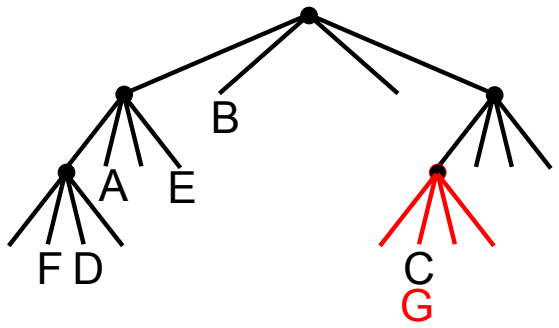
Příklad PR kvadrantového stromu



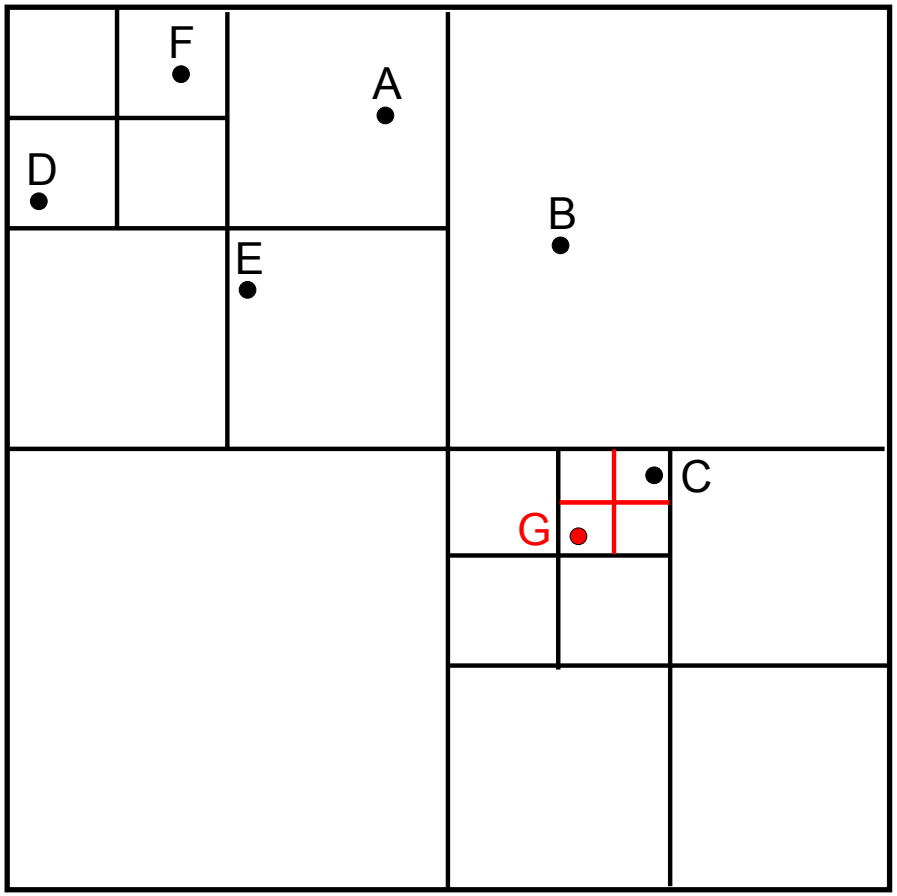


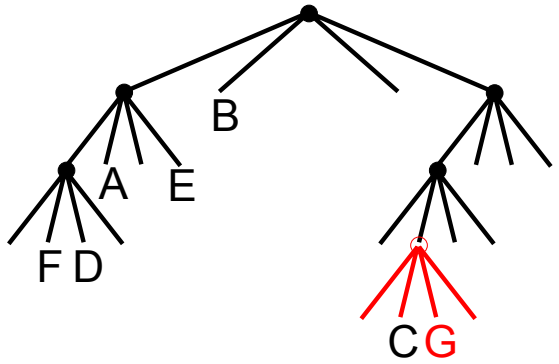
Příklad PR kvadrantového stromu



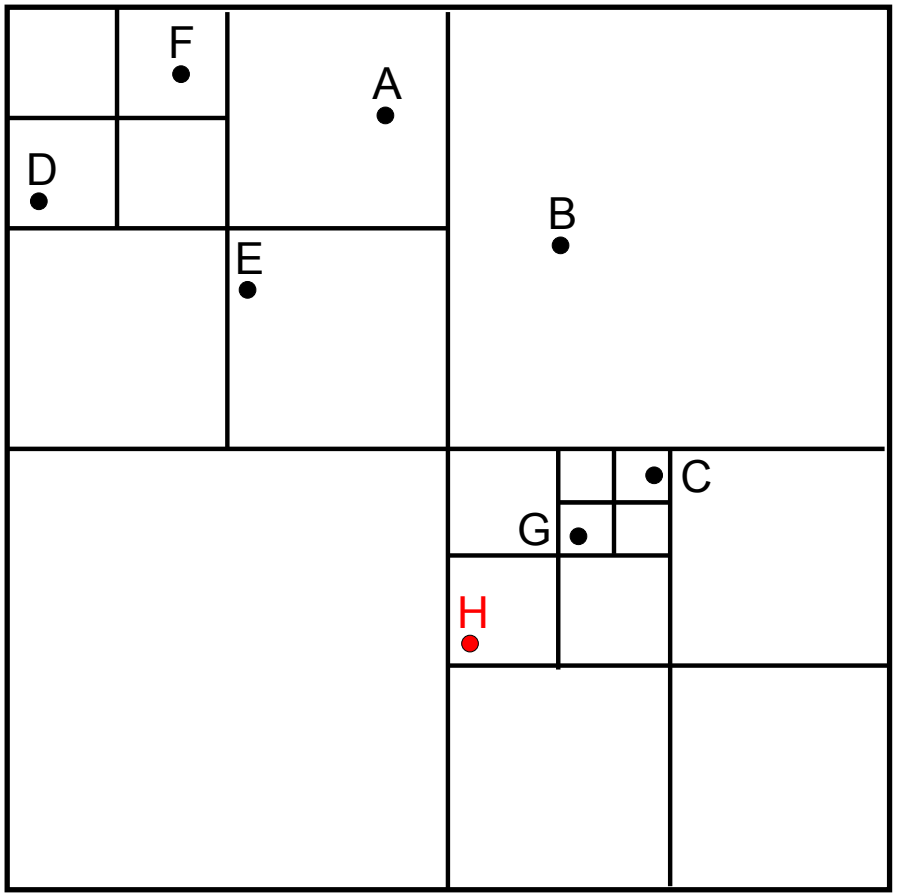


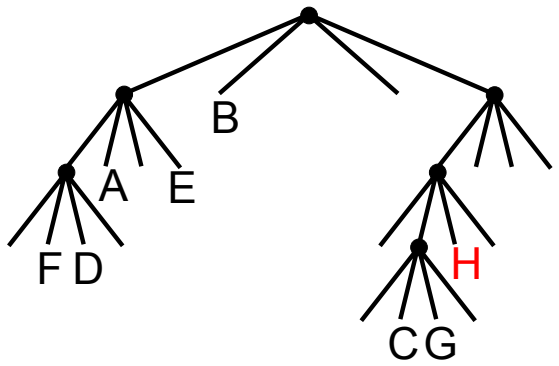
Příklad PR kvadrantového stromu



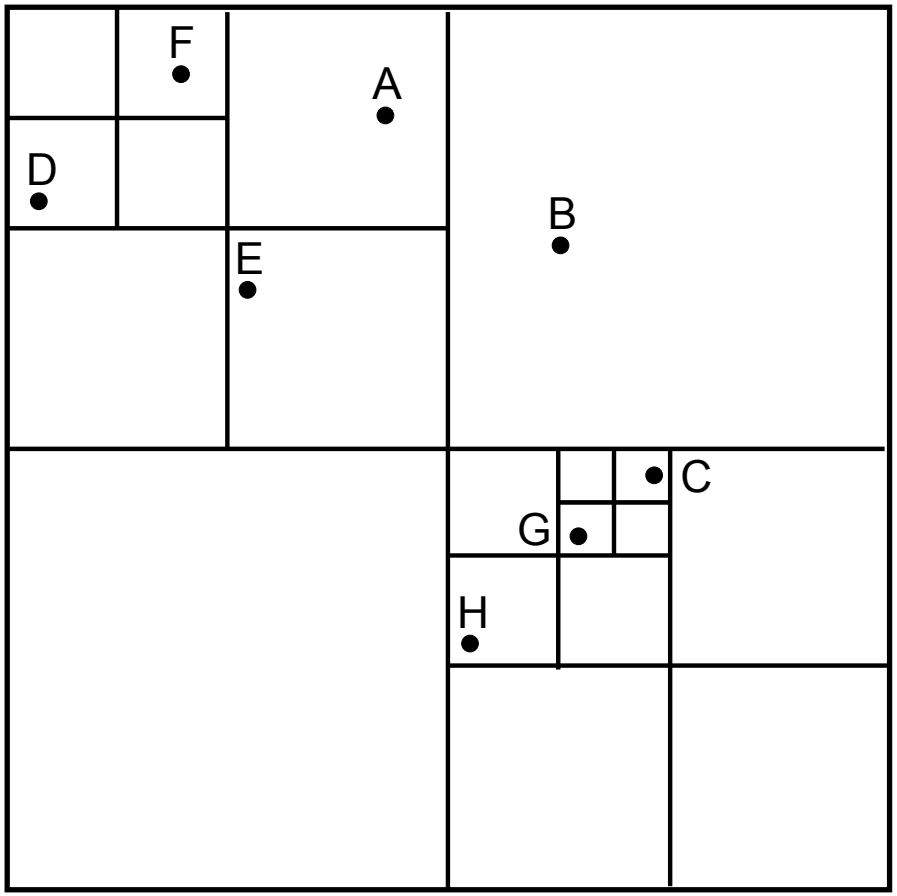


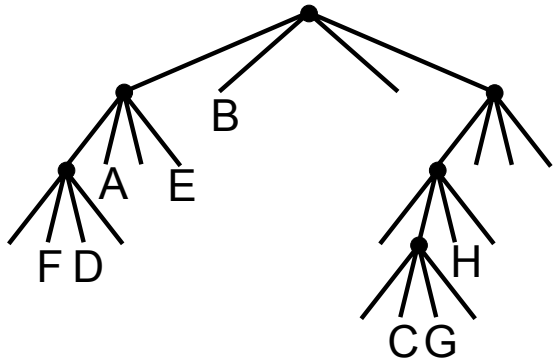
Příklad PR kvadrantového stromu





Příklad PR kvadrantového stromu





Vložení elementu do PR kvadrantového stromu

Postup pro vložení prvku,řekněme $(coord, elem)$, je velmi podobné pro vložení do BVS jen ve dvou rozměrech.

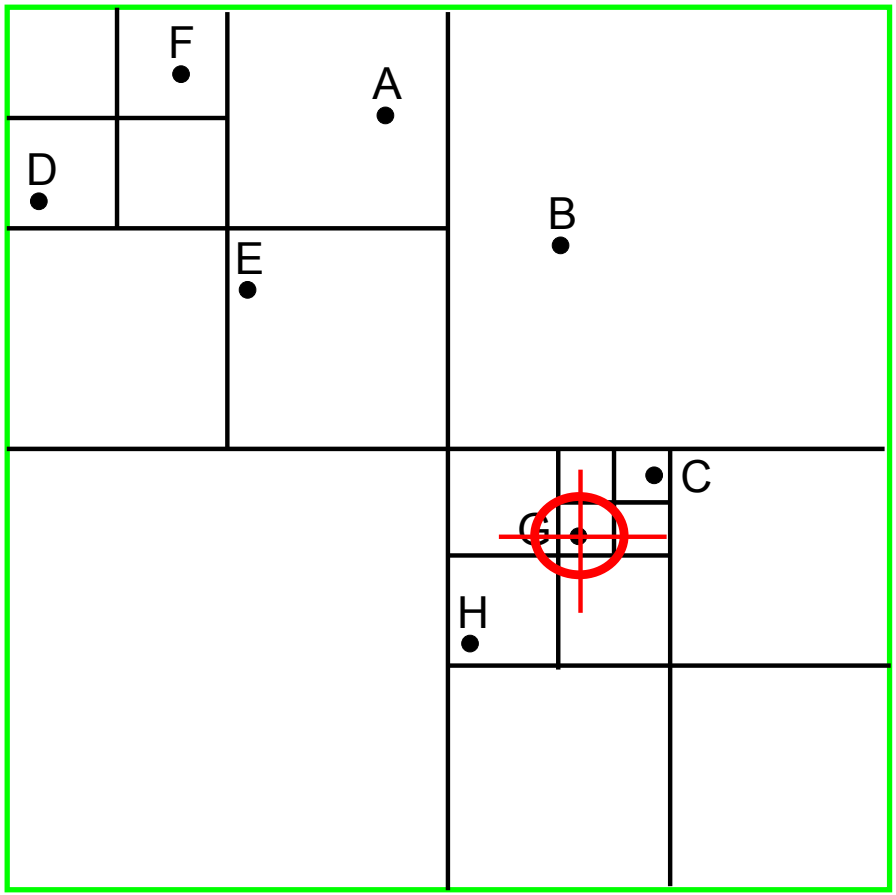
1. Nejprve se podle souřadnice nalezne list, do kterého nově vkládáný element patří
   - Klíči jsou zadané souřadnice (dvě pro 2D prostor), které porovnáváme se středem oblasti, kterou reprezentuje prohledávaný uzel a vybíráme příslušný kvadrant, do kterého souřadnice patří.
2. Pokud je list prázdný (neobsahuje žádný element), pak je element $elem$ vložen do listu.
3. Pokud je list plný, provede se dekompozice uzlu
   - Vytvoří se čtyři potomci dekomponovaného uzlu.
   - Element původně uložený v dekomponovaném uzlu se přesune do odpovídajícího syna (dle souřadnice uloženého elementu).
   - Rekursivně se provede operaci vkládání pro dekomponovaný uzel.

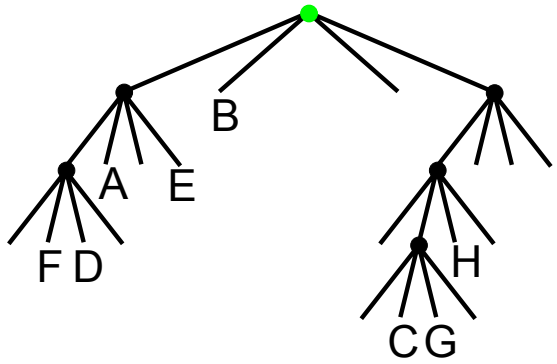
Hledání elementu v PR kvadrantovém stromu

Algoritmus vyhledání oblasti obsahující hledaný element s 2D souřadnicí $coord$ v kvadrantovém stromu s kořenem r:

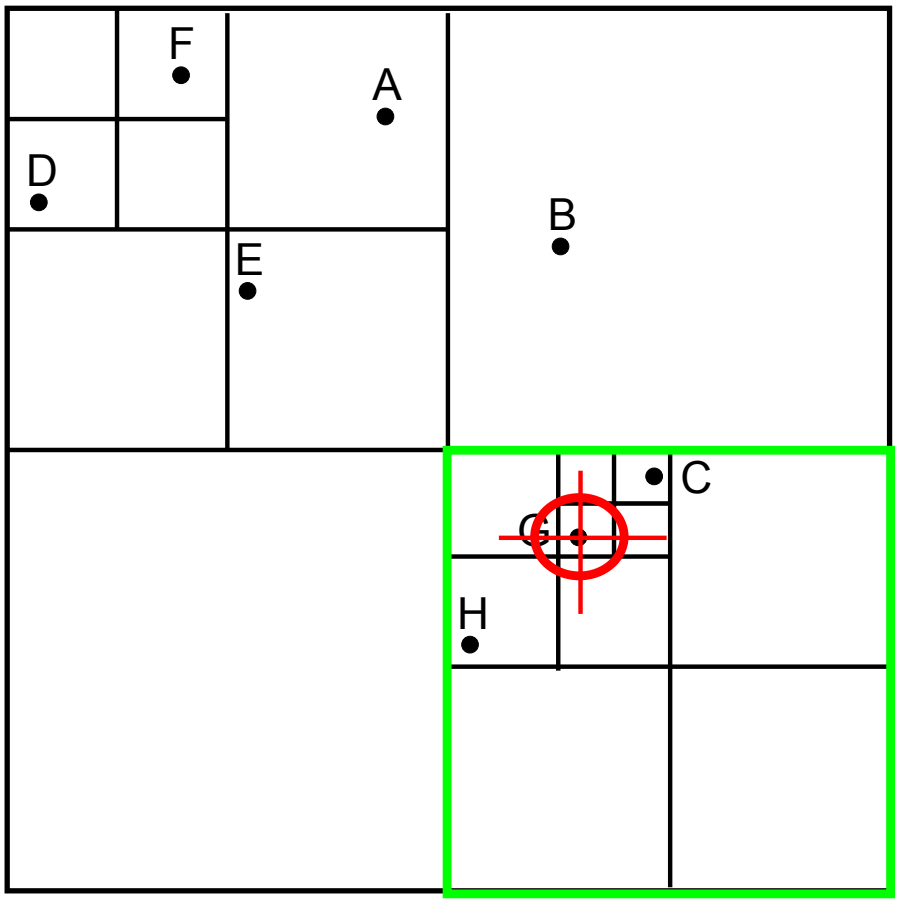
- Otestuje se, zda je kořen r listem:
  - Pokud ano, otestuje se, zda je kořen plný (obsahuje odkaz na uložený element):
    - ⋆ Pokud ano, provede se operace nad hledaným elementem a zadanou souřadnicí $coord$ (např. detekce kolize souřadnice s uloženým elementem).
    - ⋆ Pokud ne, je hledání ukončeno jako neúspěšné.
  - Pokud ne, provede se rekursivně totéž pro odpovídající podstrom (dle hledané souřadnice $coord$) reprezentující jeden z kvadrantů.

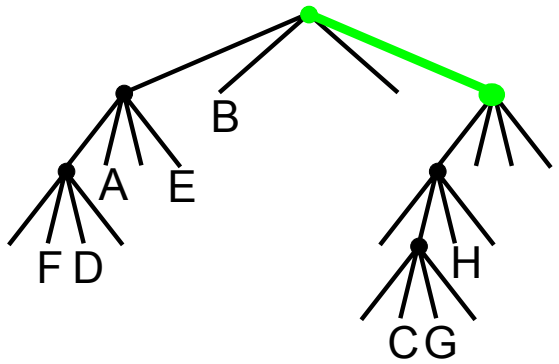
Hledání v PR kvadrantového stromu



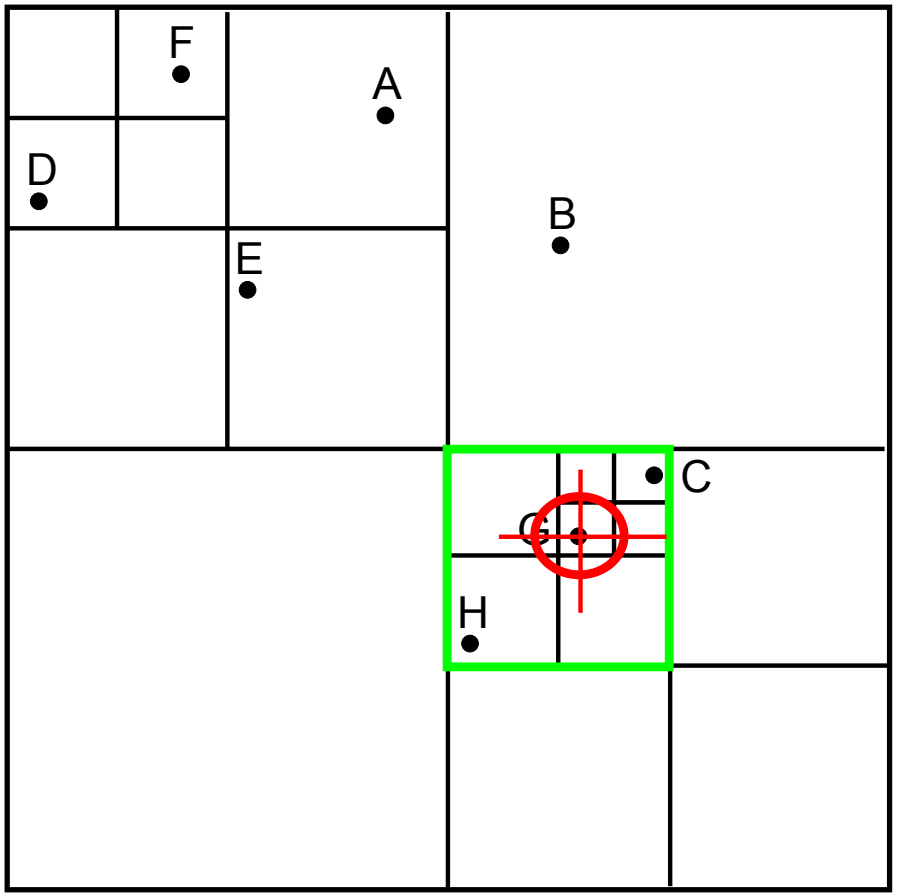


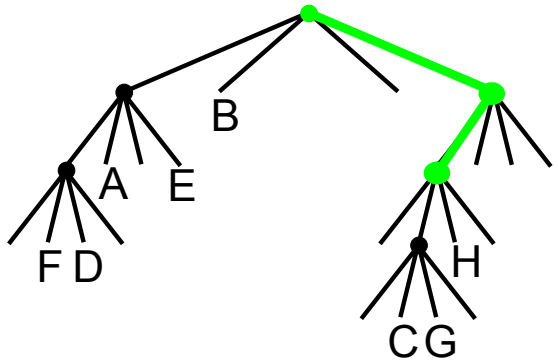
Hledání v PR kvadrantového stromu



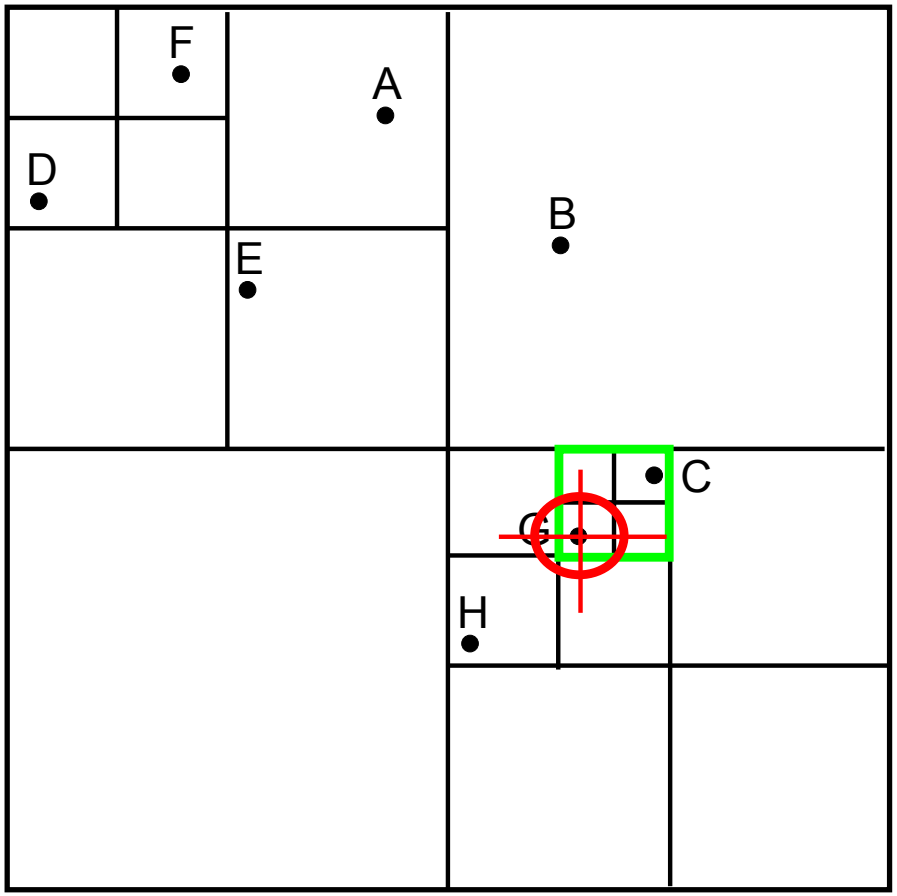


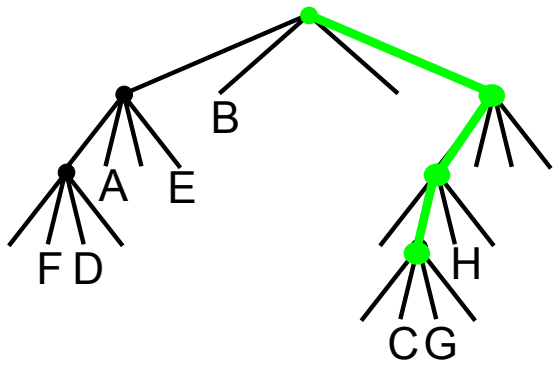
Hledání v PR kvadrantového stromu



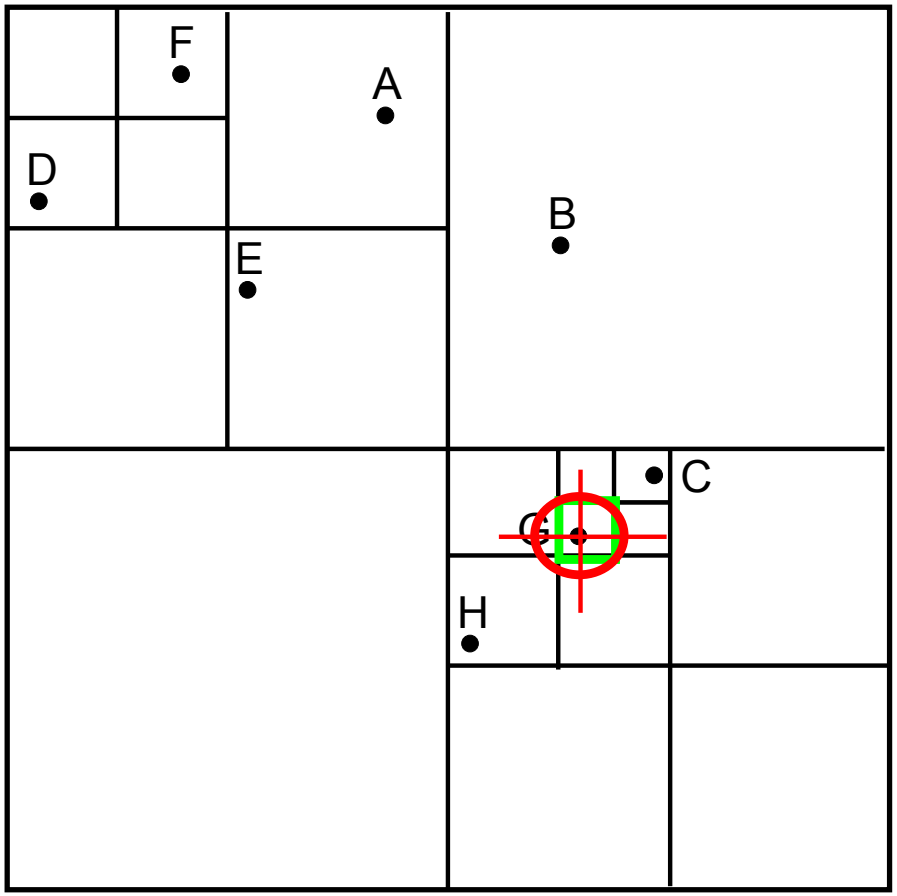


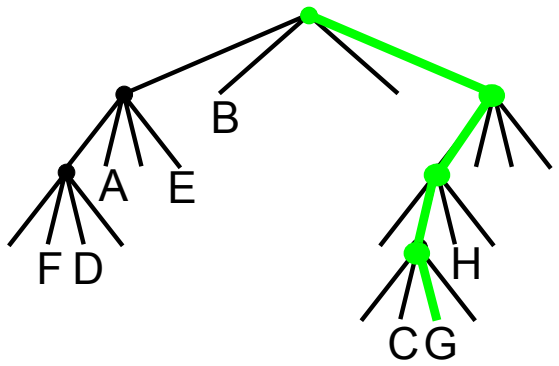
Hledání v PR kvadrantového stromu





Hledání v PR kvadrantového stromu

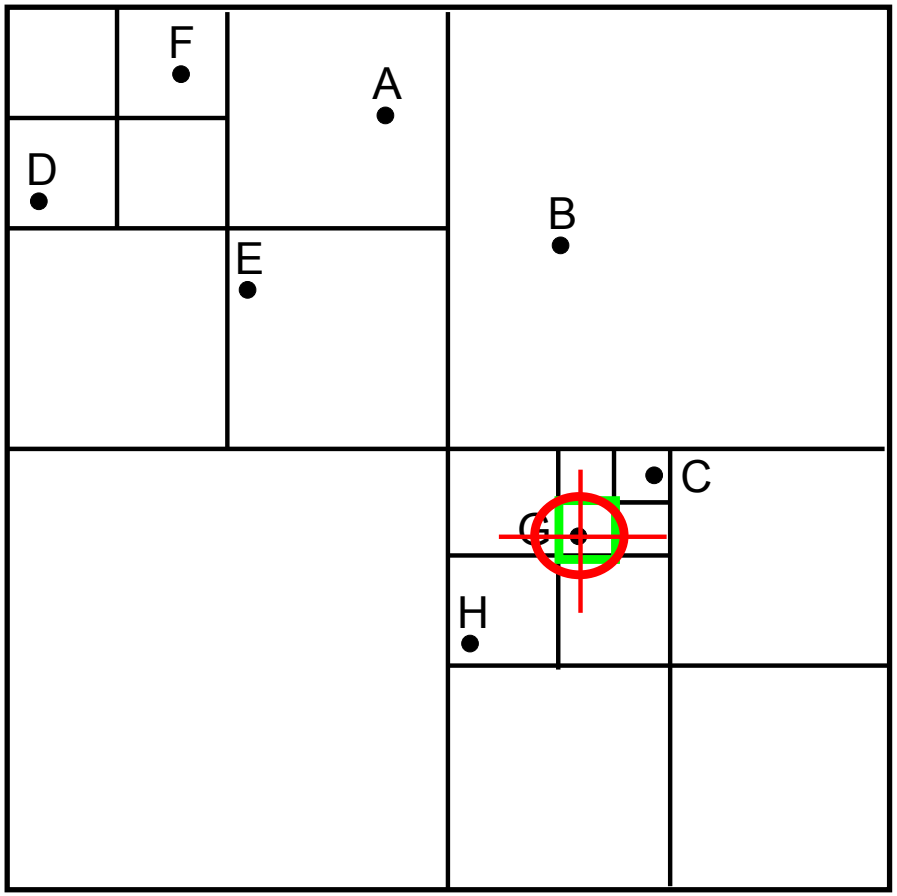


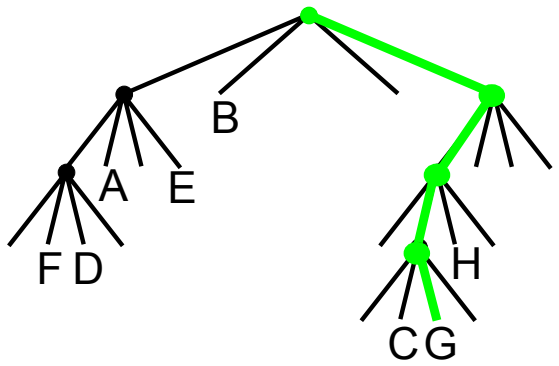


Mazání elementu z PR kvadrantového stromu

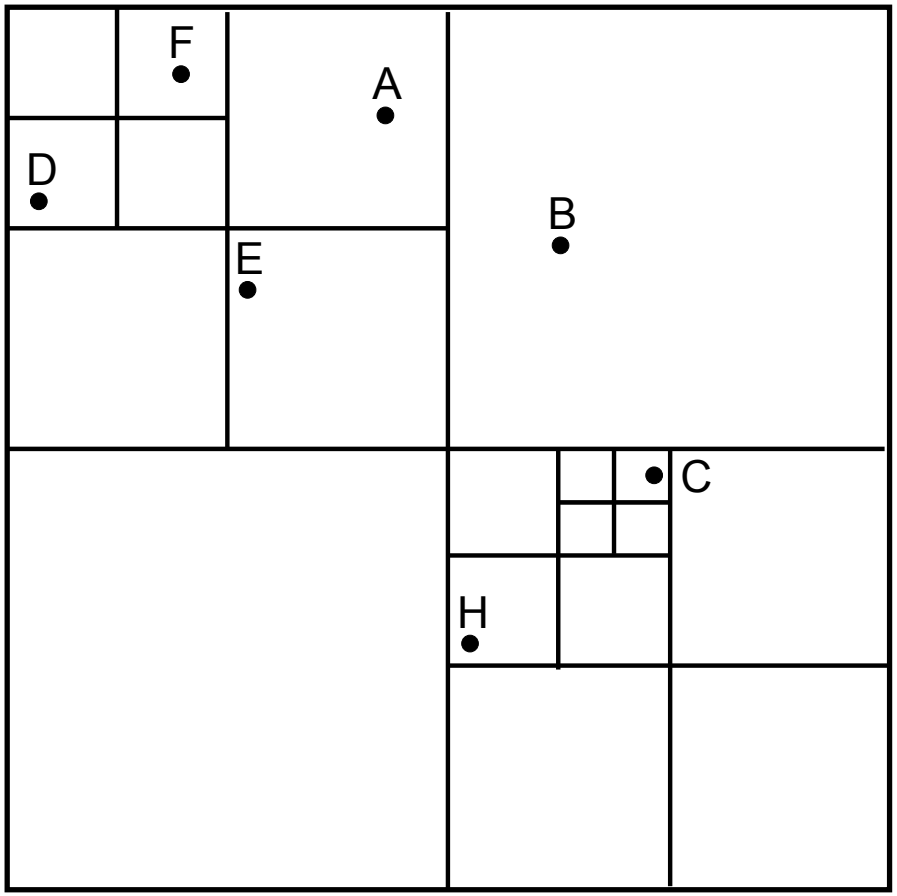
1. Nalezne se list, který obsahuje hledaný element.
2. Provede se výmaz daného elementu.
3. Otestují se sourozence listu, zda jsou listy a jsou prázdné:
   - Pokud ano, stane se jejich otec listem. A vracím se na bod 3.
   - Pokud ne, konec mazání.

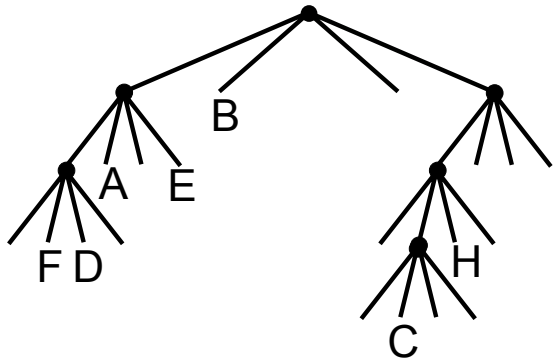
Mazání v PR kvadrantového stromu



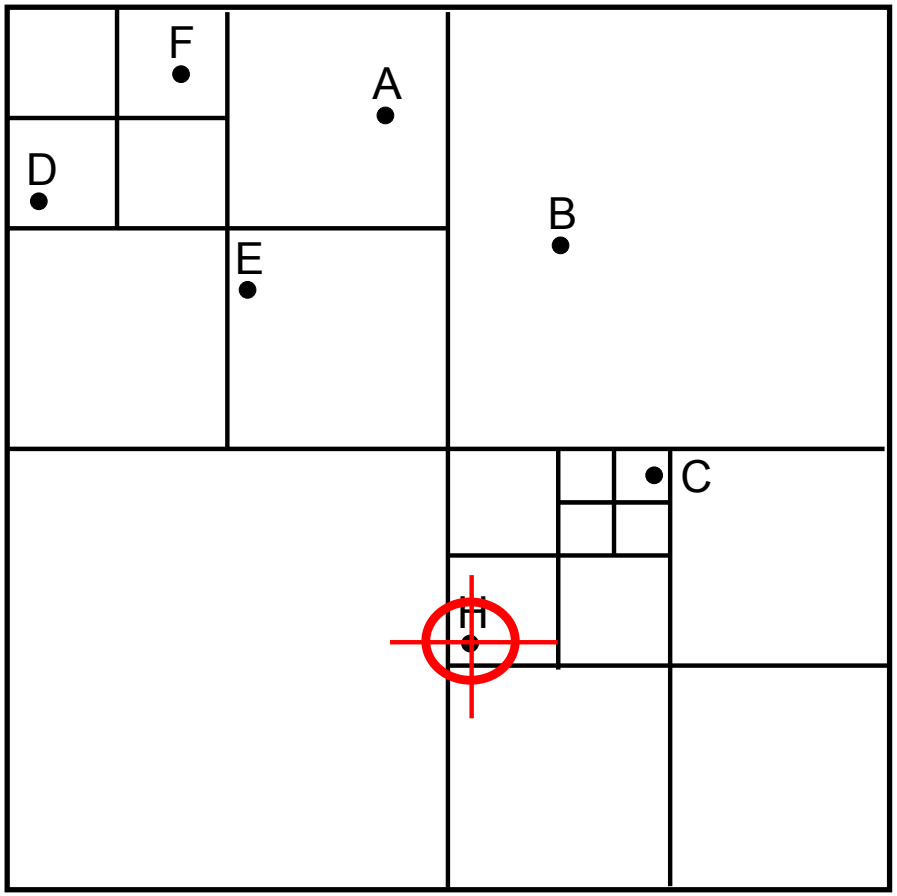


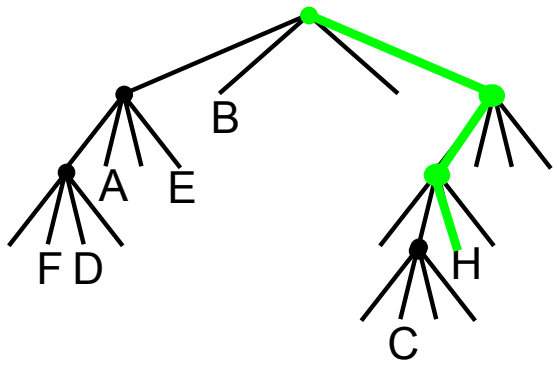
Mazání v PR kvadrantového stromu



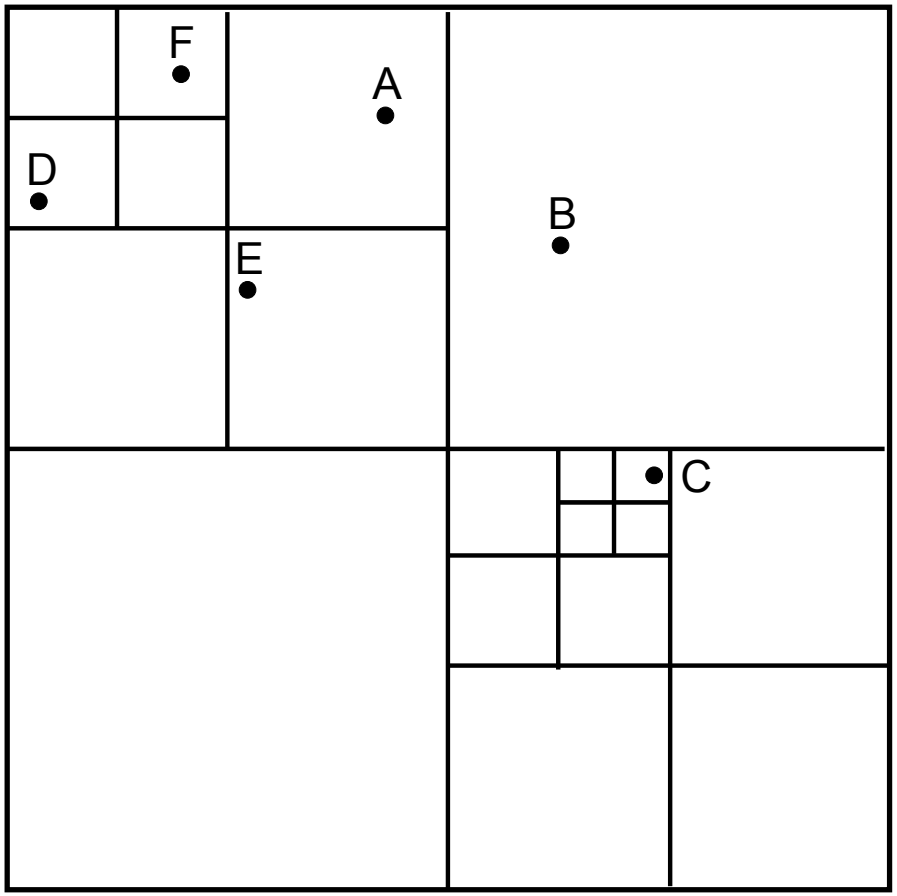


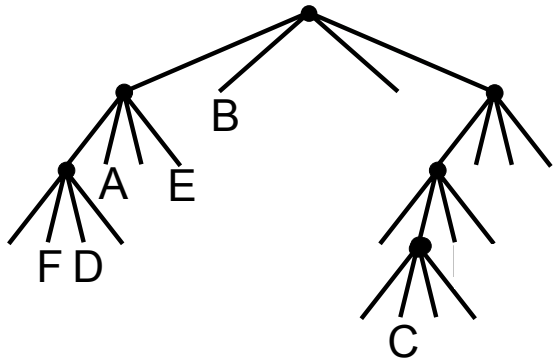
Mazání v PR kvadrantového stromu



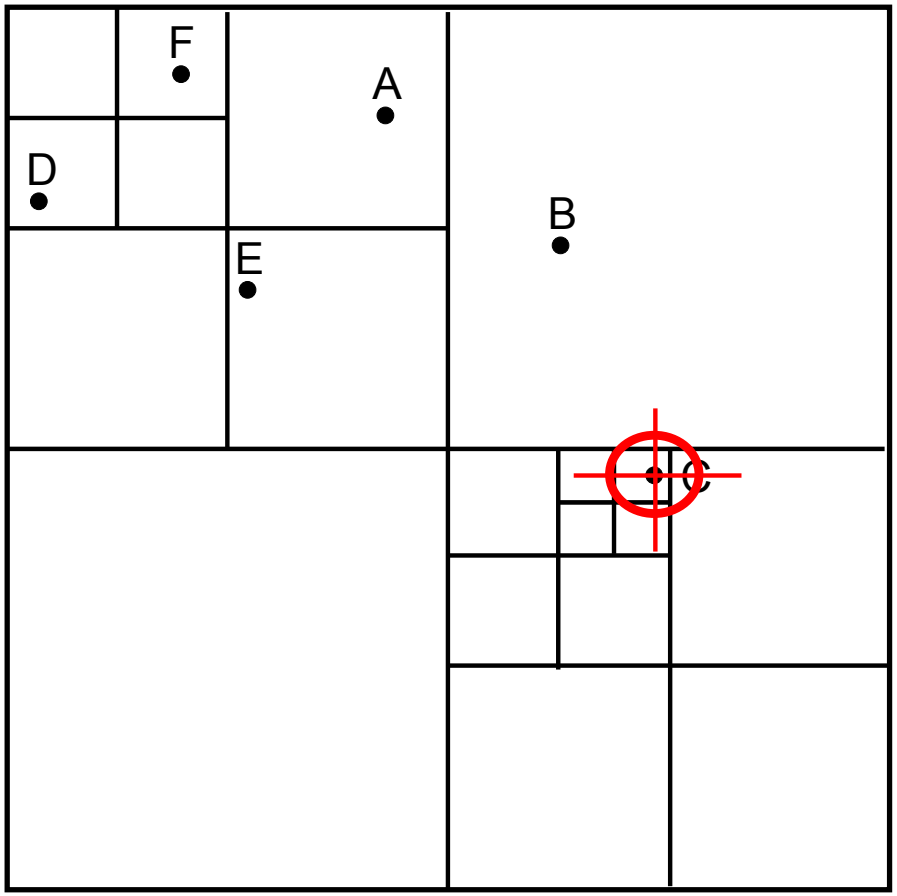


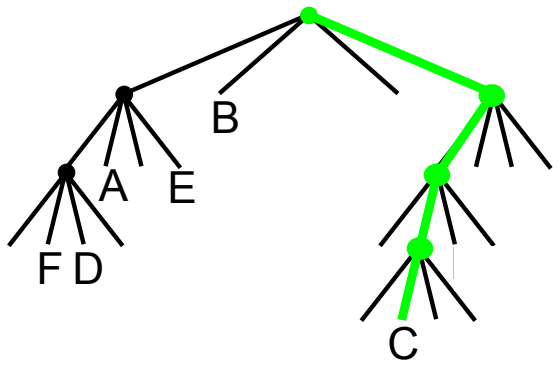
Mazání v PR kvadrantového stromu



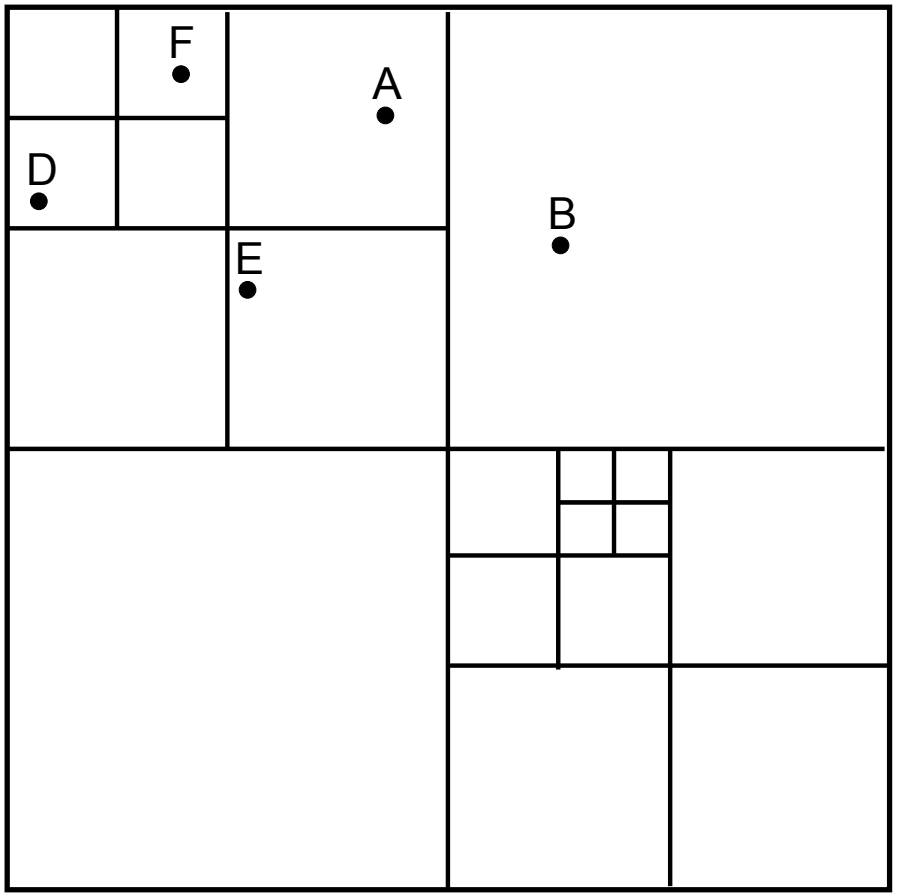


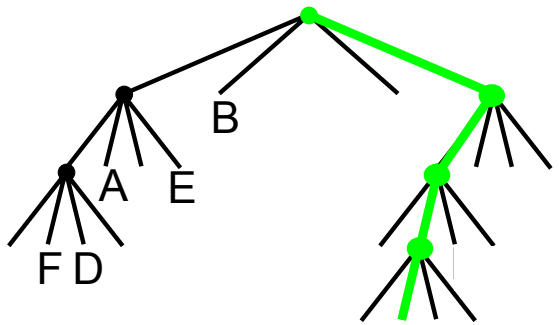
Mazání v PR kvadrantového stromu



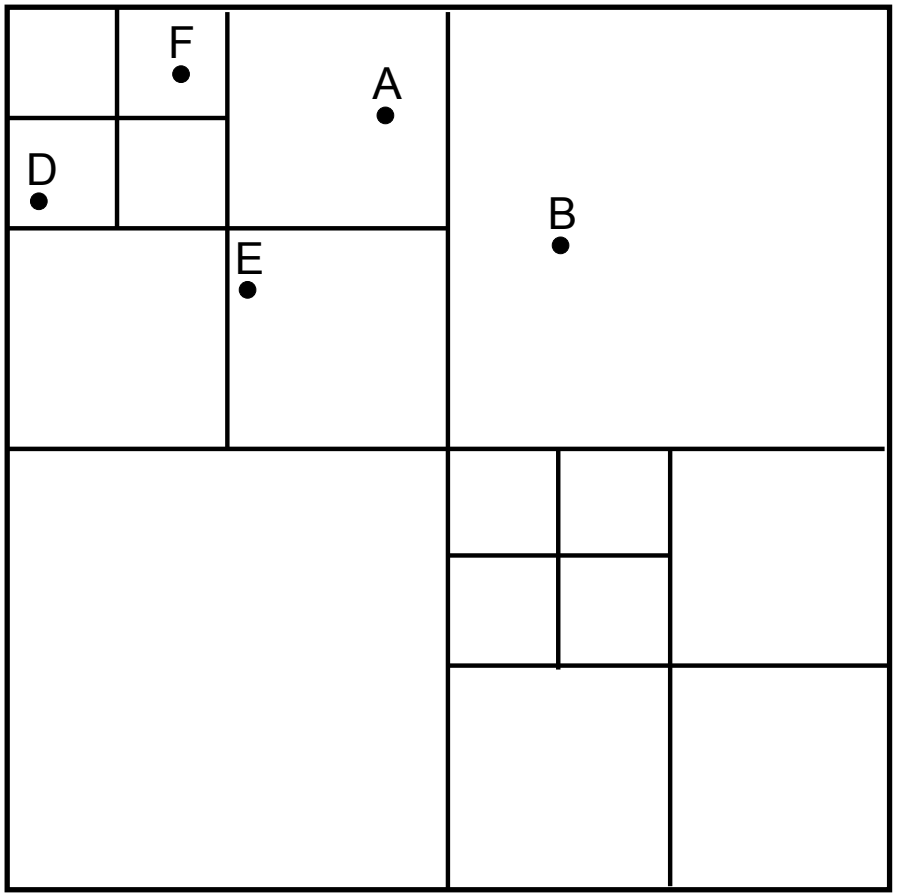


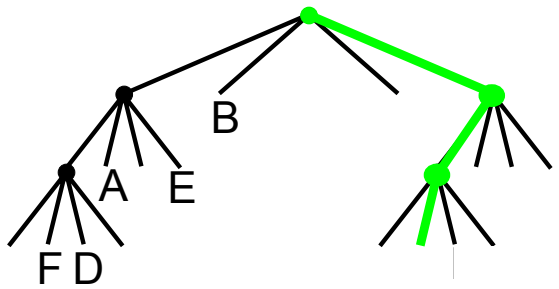
Mazání v PR kvadrantového stromu



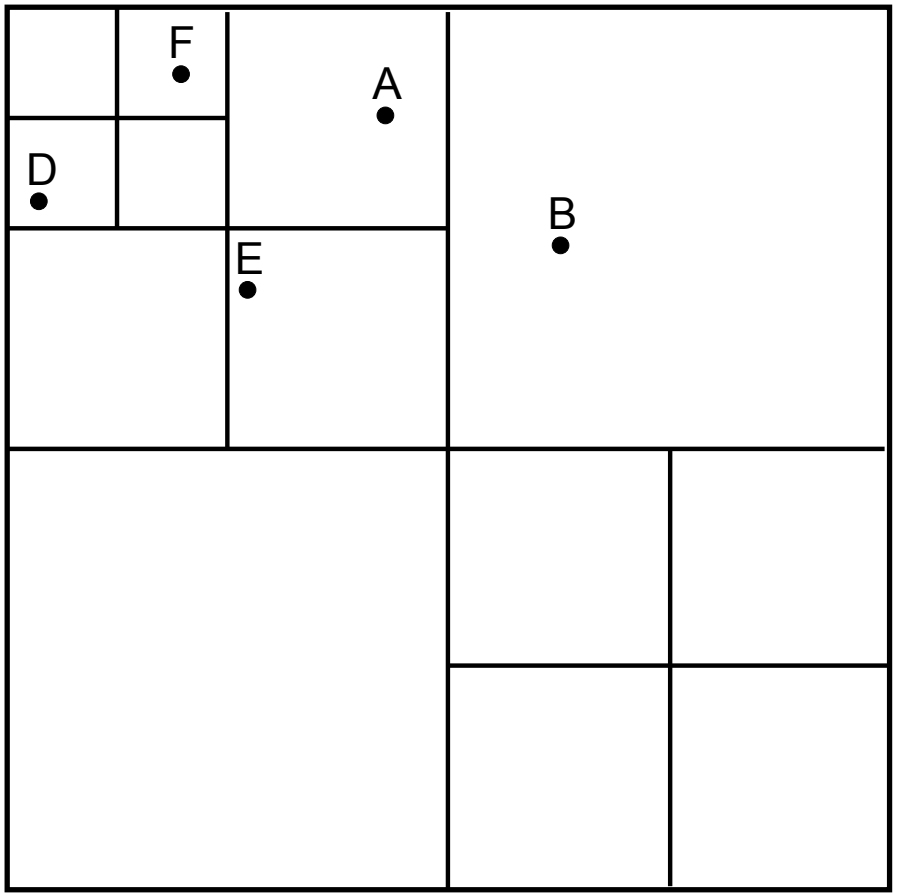


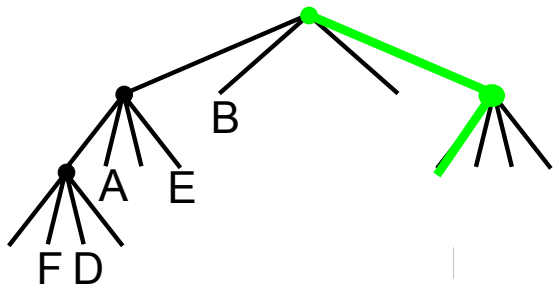
Mazání v PR kvadrantového stromu



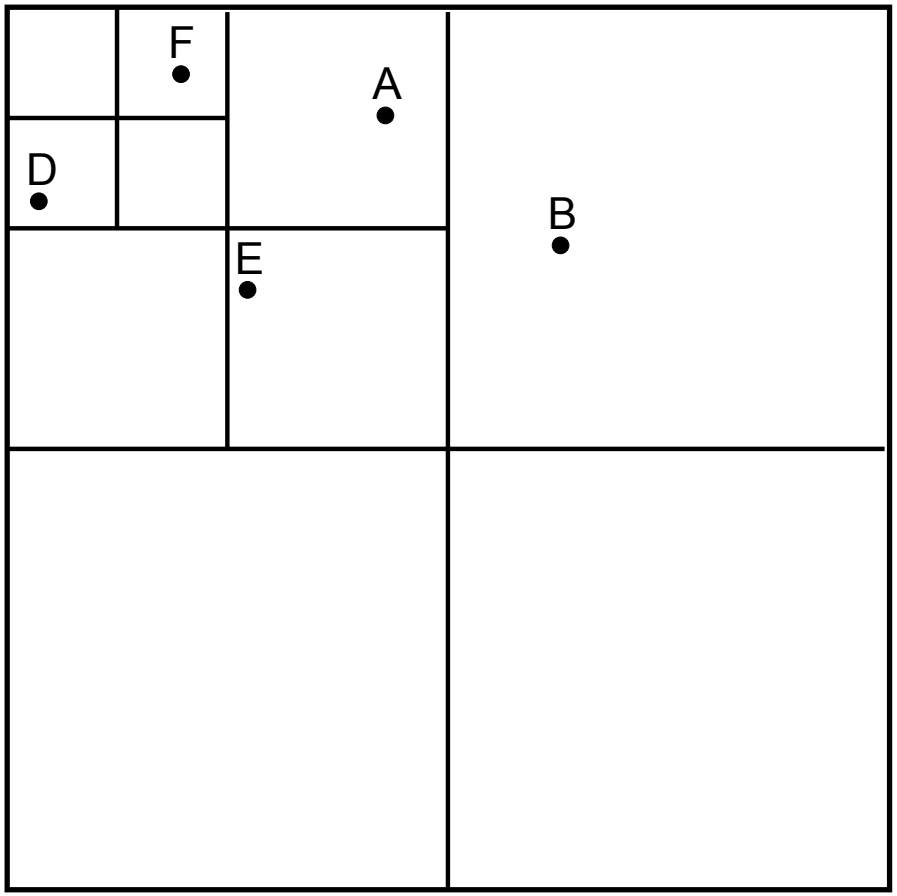


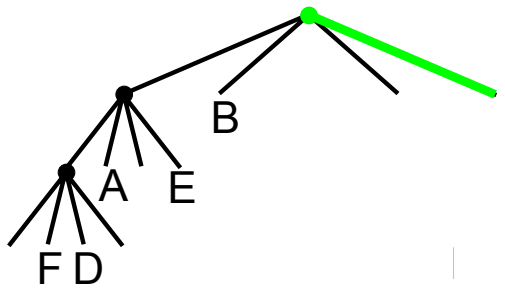
Mazání v PR kvadrantového stromu



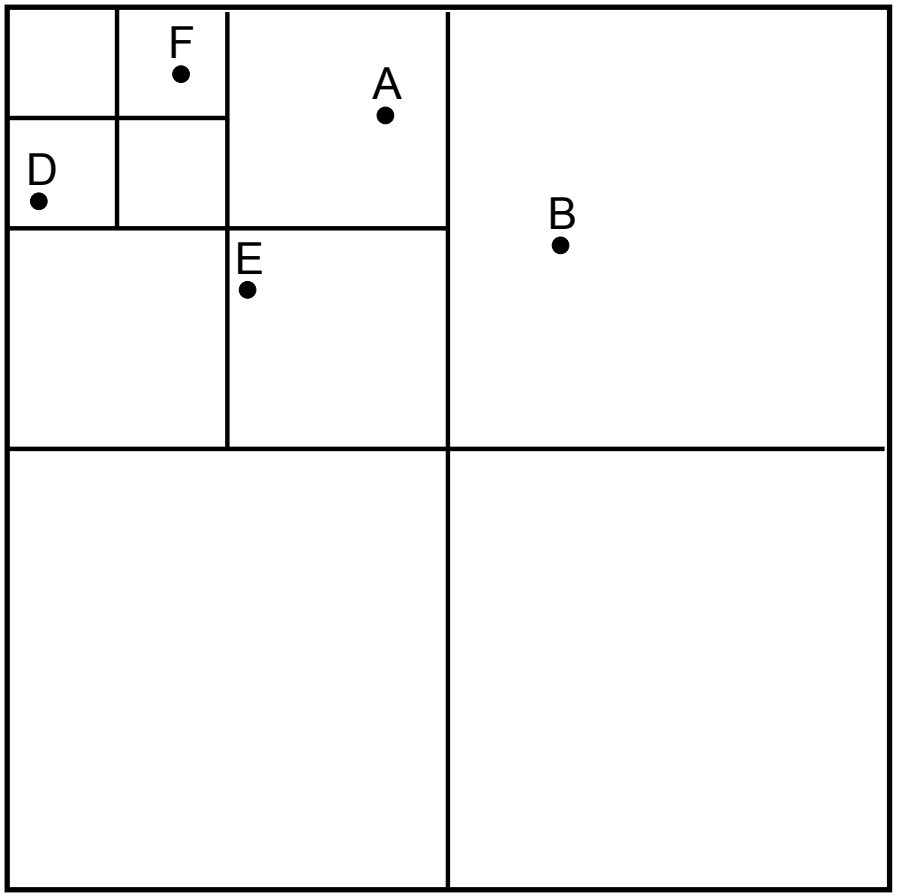


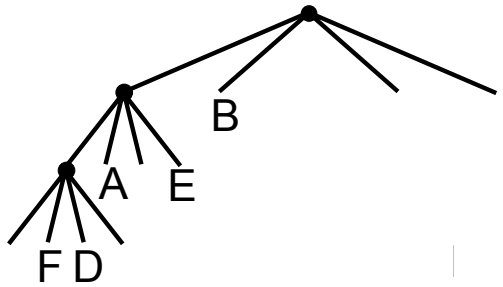
Mazání v PR kvadrantového stromu





Mazání v PR kvadrantového stromu





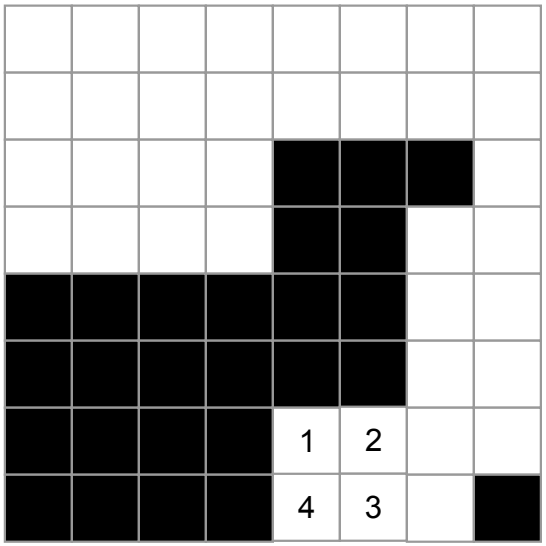
Hodnocení PR kvadrantového stromu

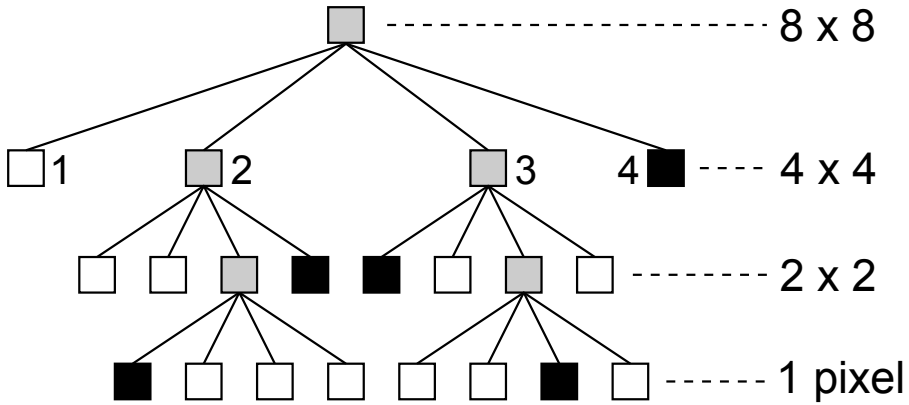
1. Struktura PR kvadrantového stromu je nezávislá na pořadí vkládaných elementů.
2. V případě větších shluků elementu podstatně roste hloubka strom.
3. V případě nerovnoměrného rozložení elementů v prostoru je strom nevyvážený.
4. Operace mazání je velmi jednoduchá.

R Kvadrantový strom (Region quadtree)

R Kvadrantový strom (Region quadtree)

- Strom reprezentuje černobílý bitmapový obrázek o velikosti $2^n \times 2^n$ pixelů.
- V listech stromu se uchovává hodnota B nebo W.
- Použití:
  - bezeztrátová komprese,
  - hledání fraktálních artefaktů.





P kvadrantový strom (Point quadtree)

Definice (P kvadrantový strom)

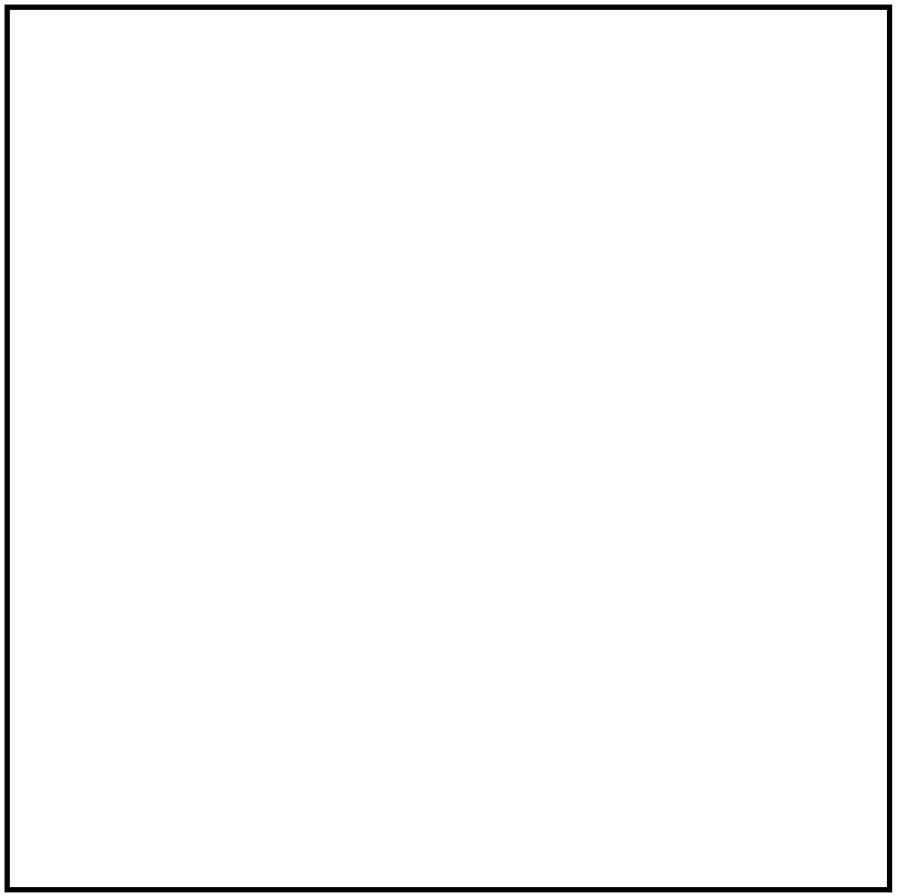
Je kvadrantový strom s těmito vlastnosti:

1. dekomponované oblasti mohou mít různou velikost, ale plně pokrývají oblast,
2. maximální kapacita pro obsažené elementy je obvykle 1,
3. uživatelská data jsou elementy a jsou uloženy ve všech uzlech,
4. souřadnice elementu ve vnitřním uzlu určuje bod rozdělení na čtyři oblasti

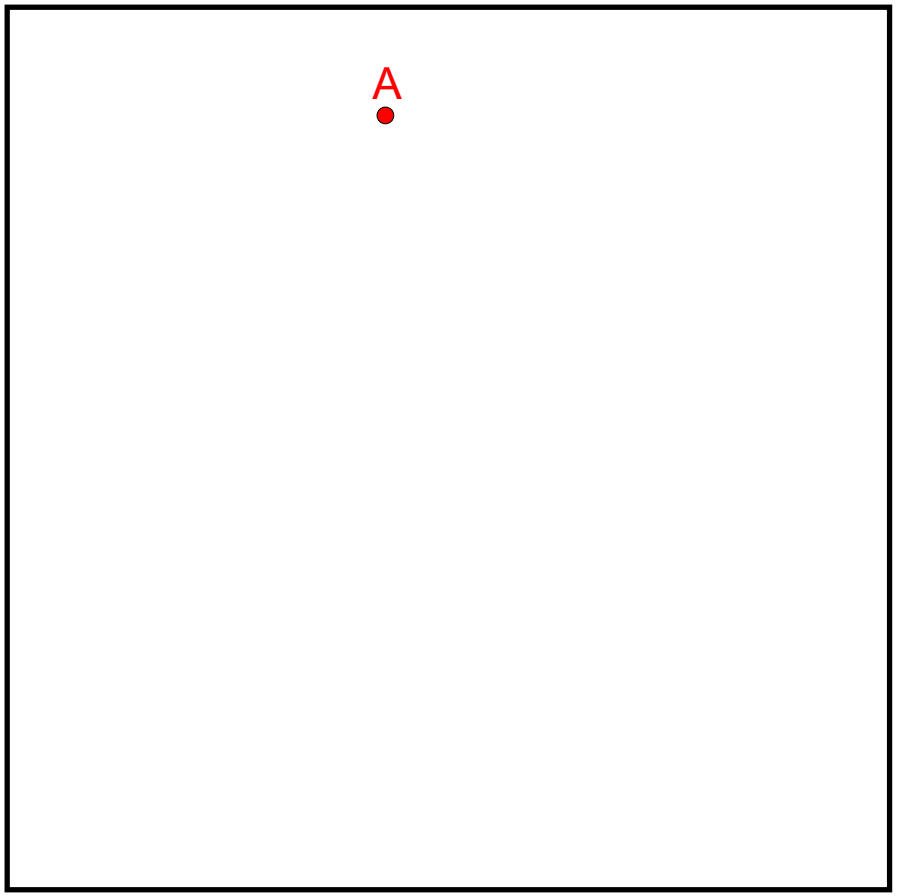
Použití:

- Obdobné jako u PR kvadrantových stromů.

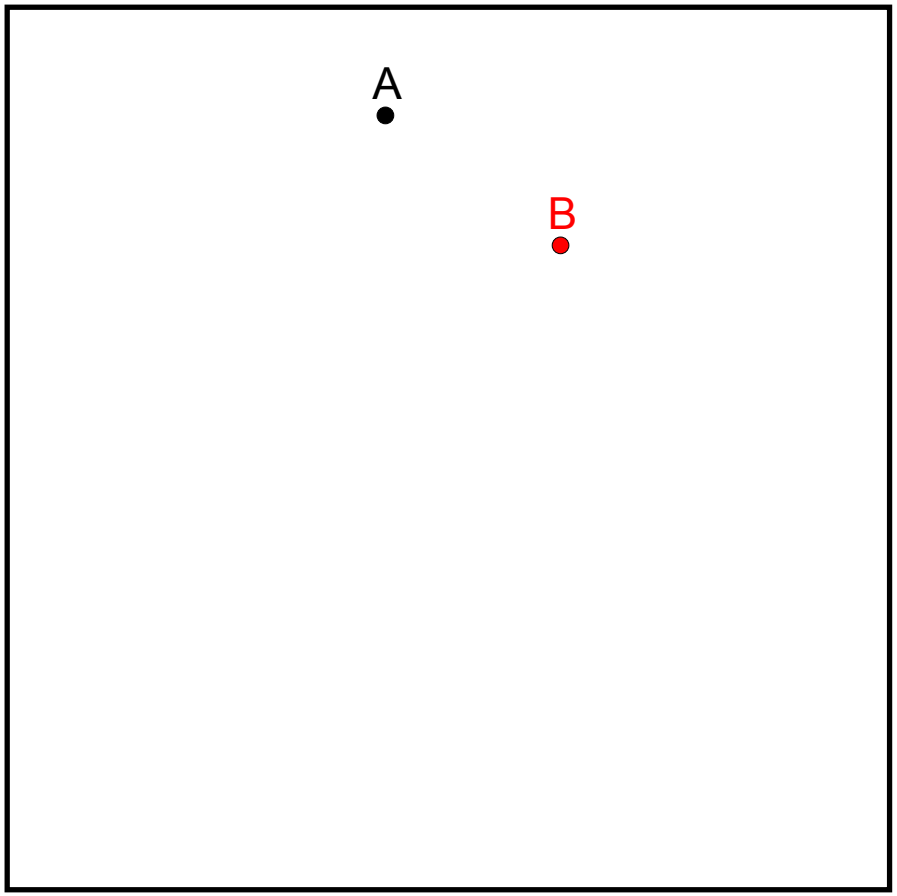
Příklad P kvadrantového stromu



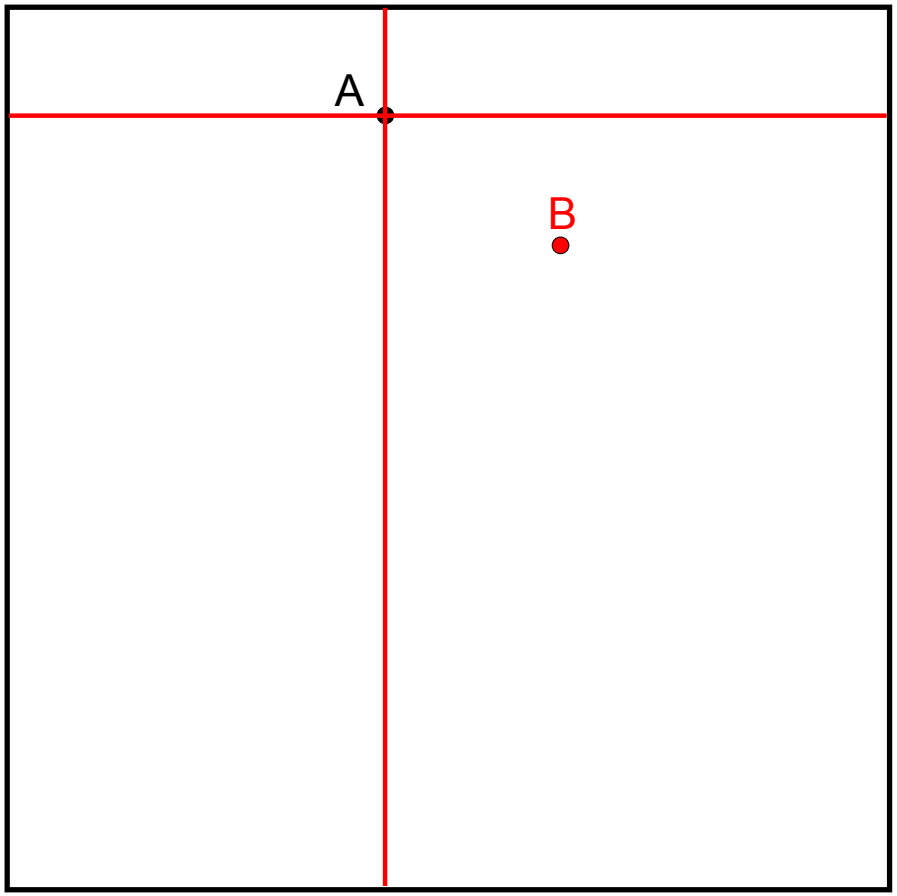
Příklad P kvadrantového stromu

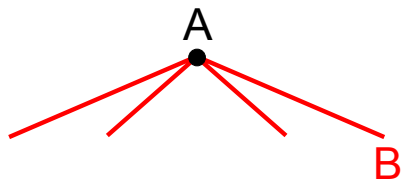


Příklad P kvadrantového stromu

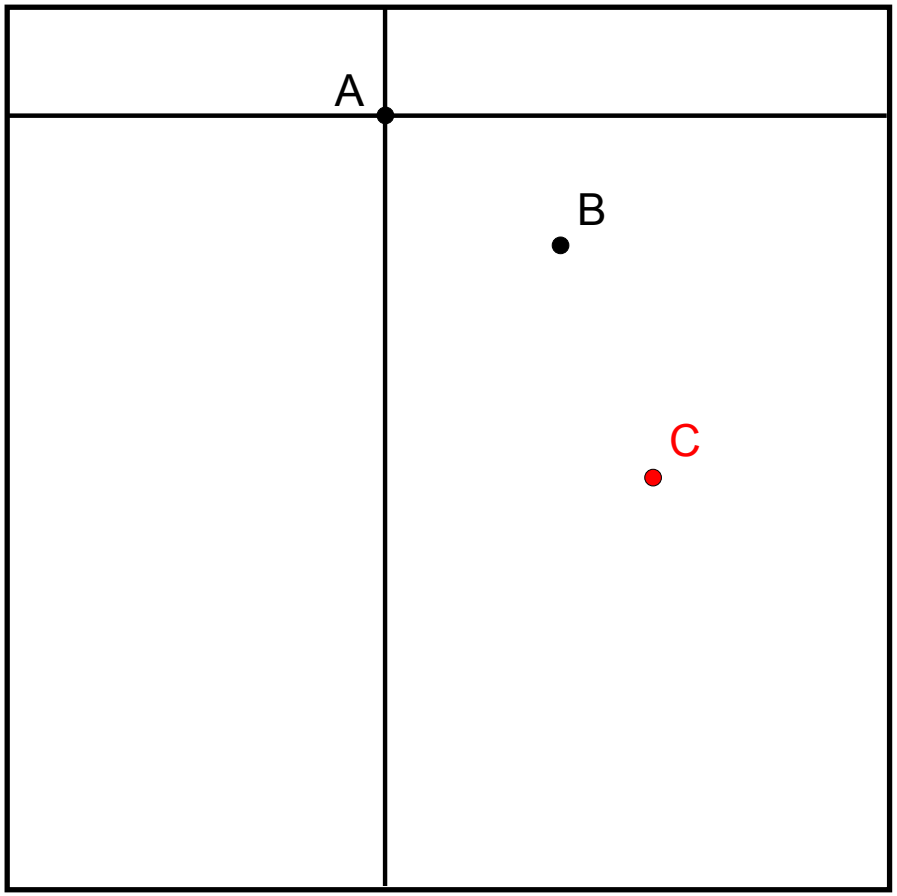


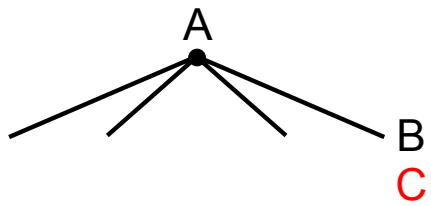
Příklad P kvadrantového stromu



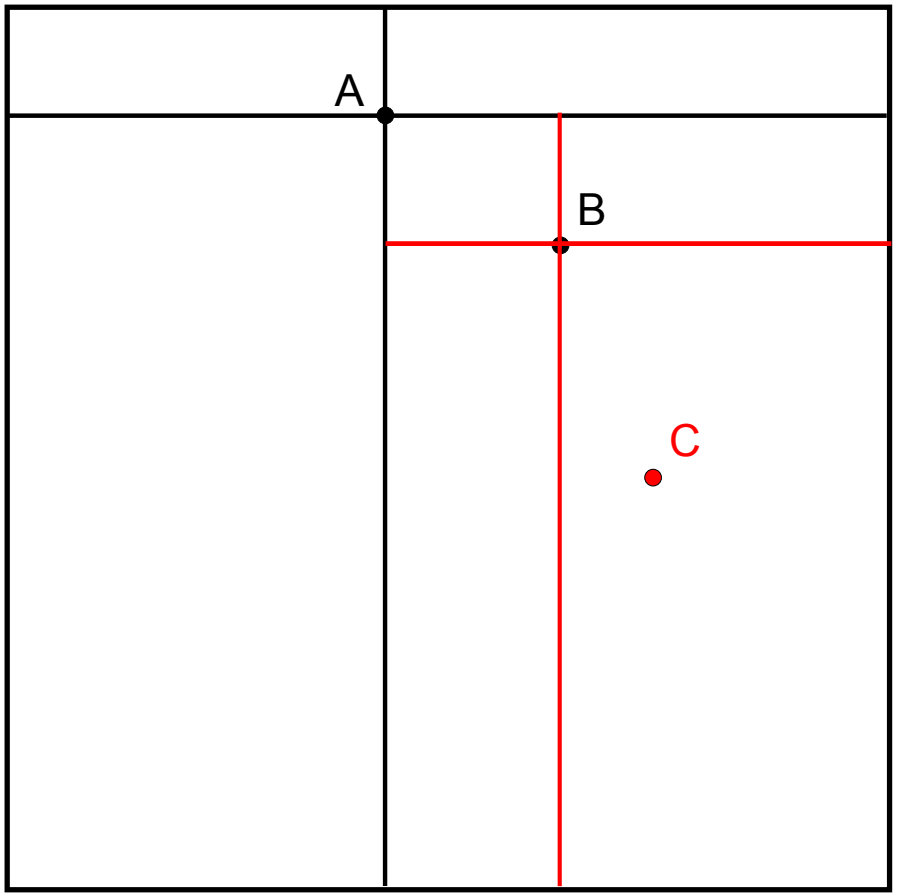


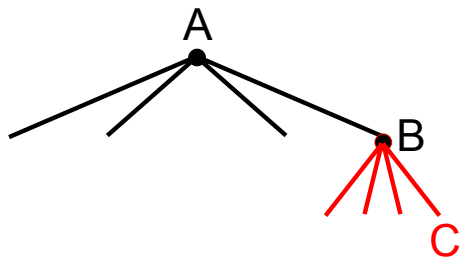
Příklad P kvadrantového stromu



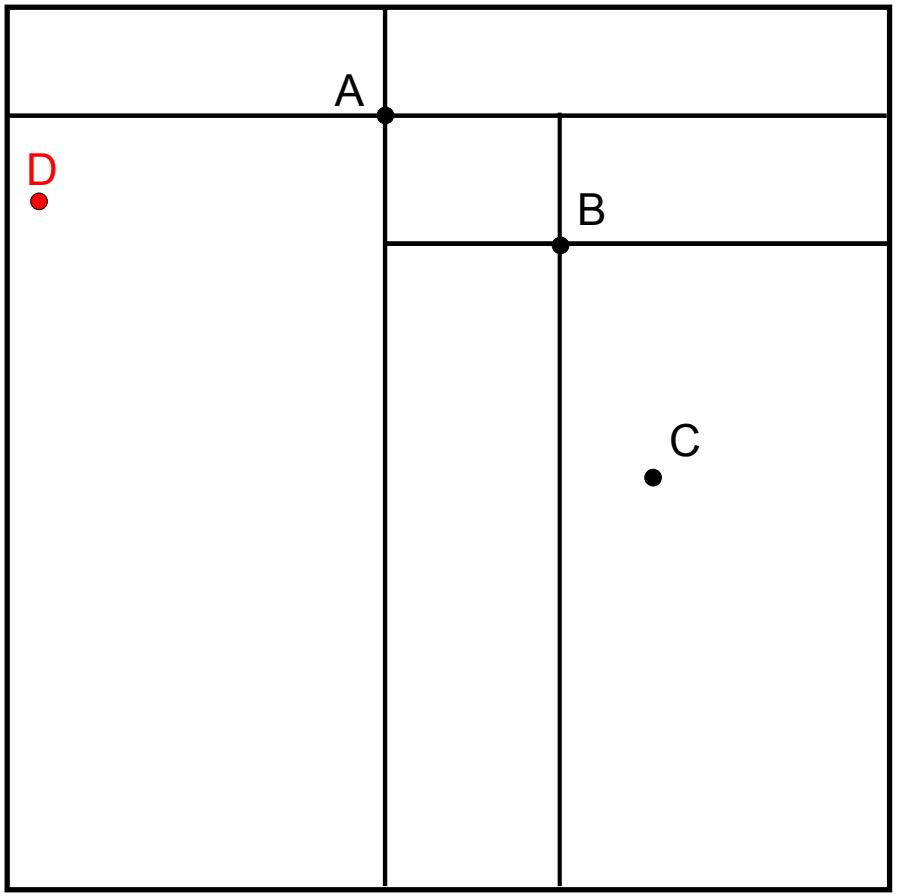


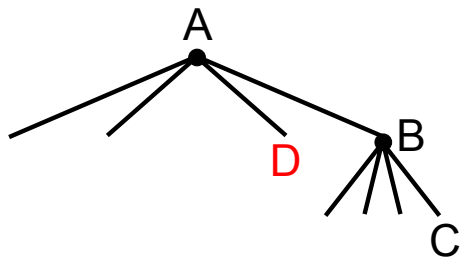
Příklad P kvadrantového stromu



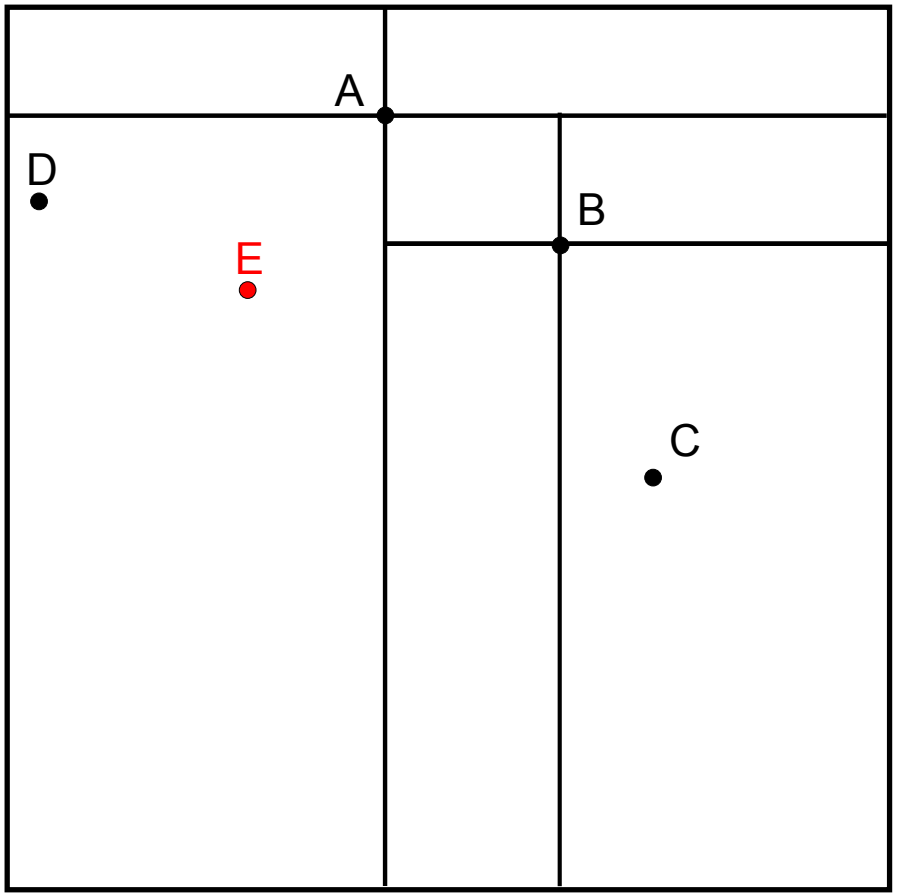


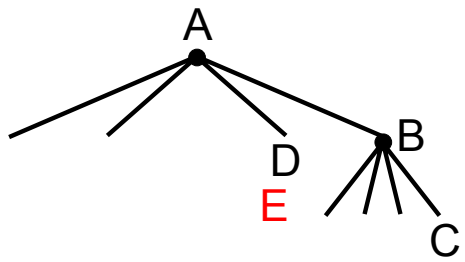
Příklad P kvadrantového stromu



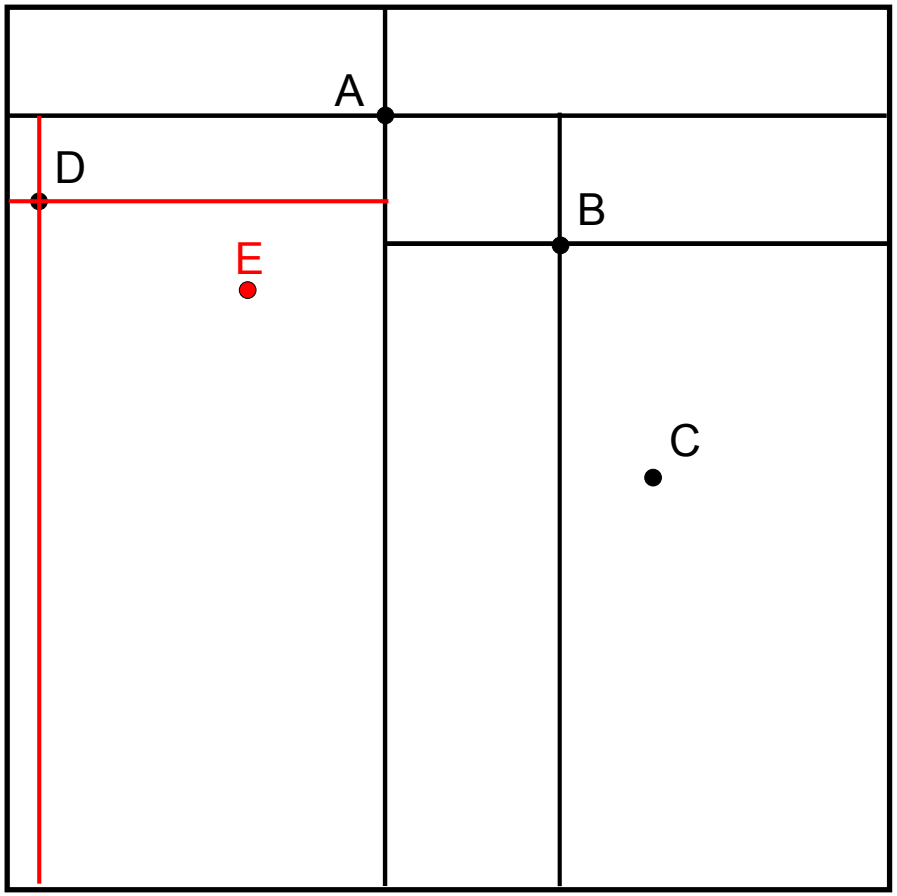


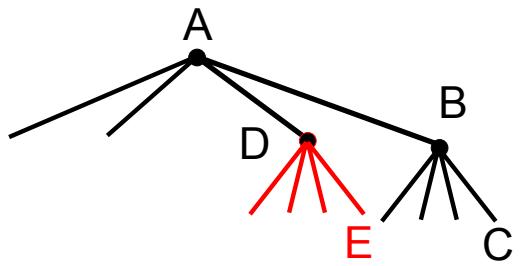
Příklad P kvadrantového stromu



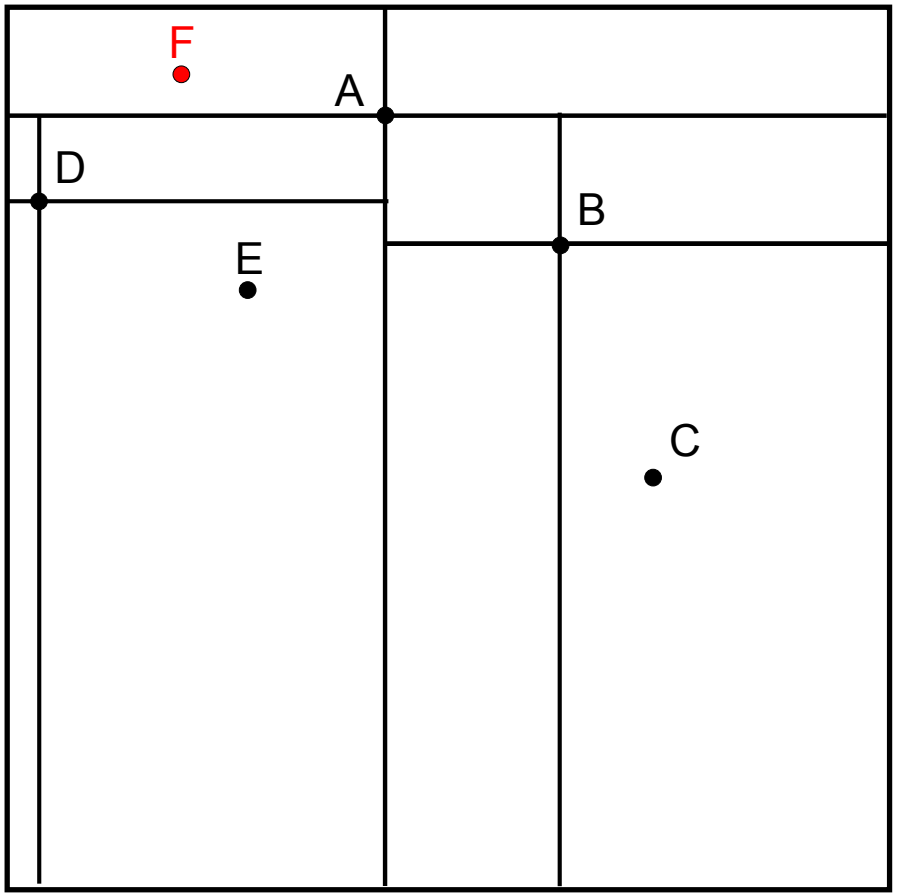


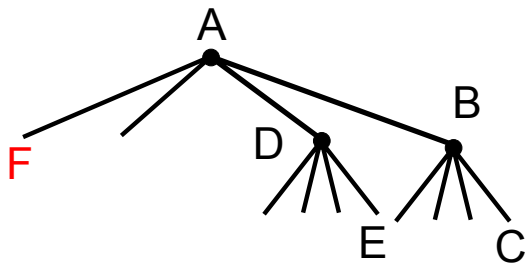
Příklad P kvadrantového stromu



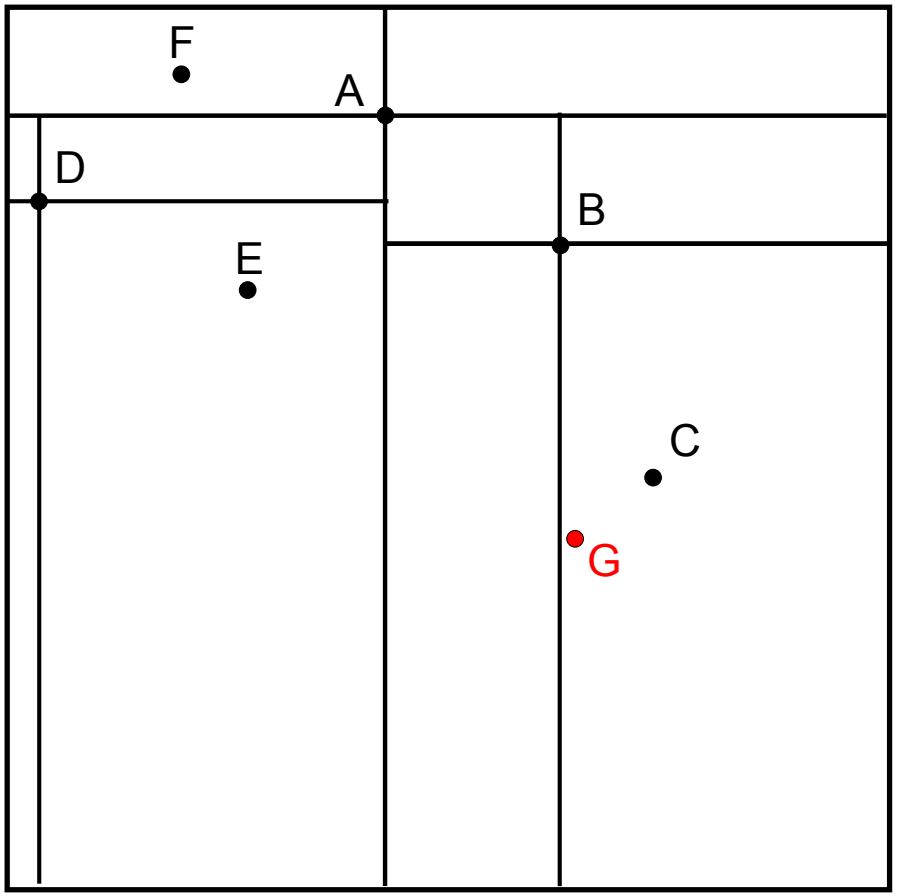


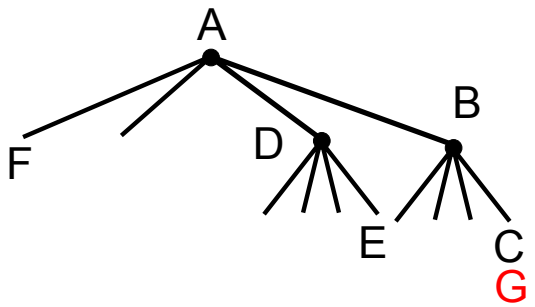
Příklad P kvadrantového stromu



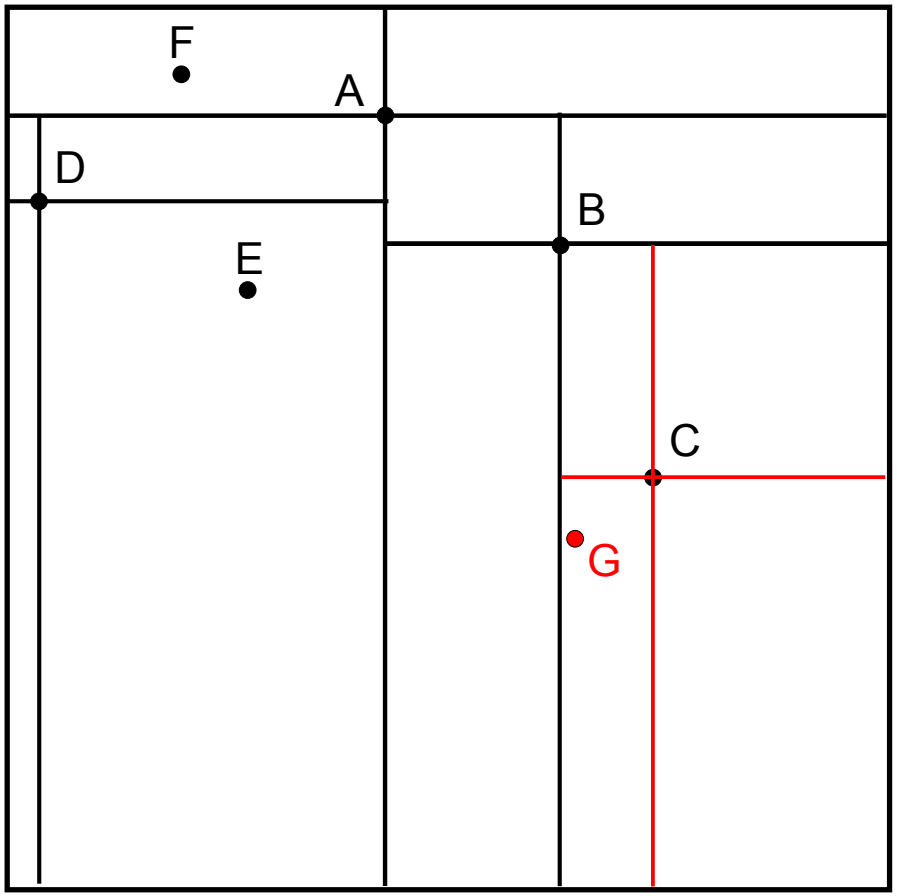


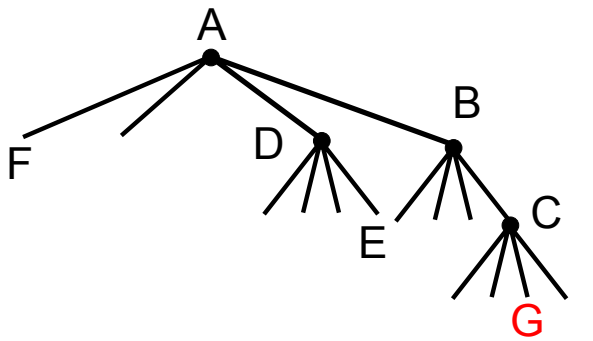
Příklad P kvadrantového stromu



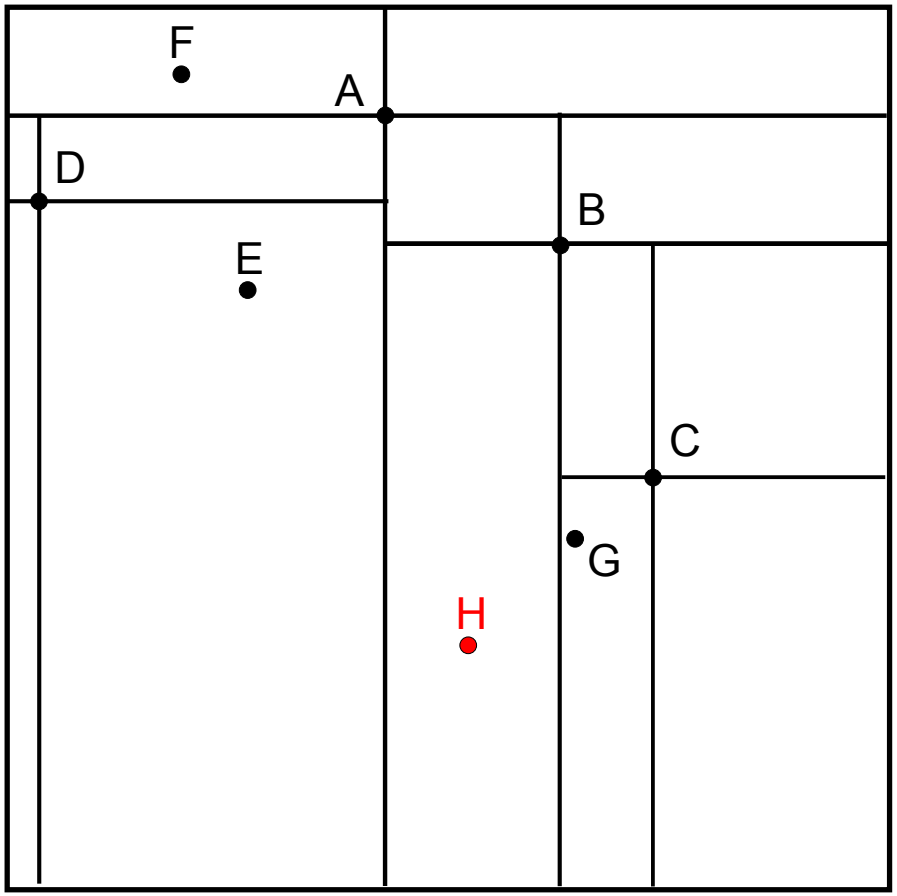


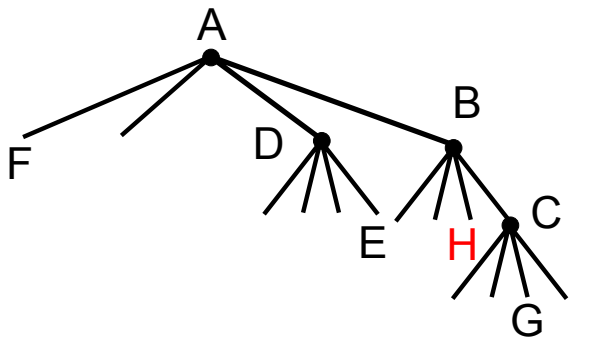
Příklad P kvadrantového stromu



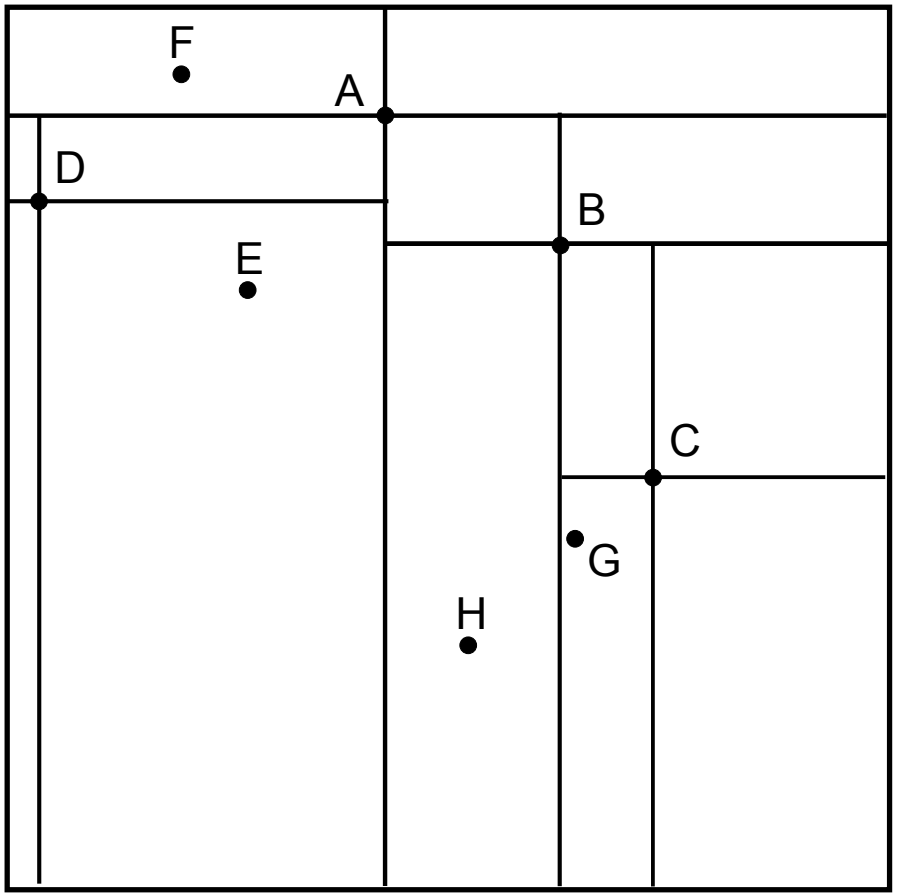


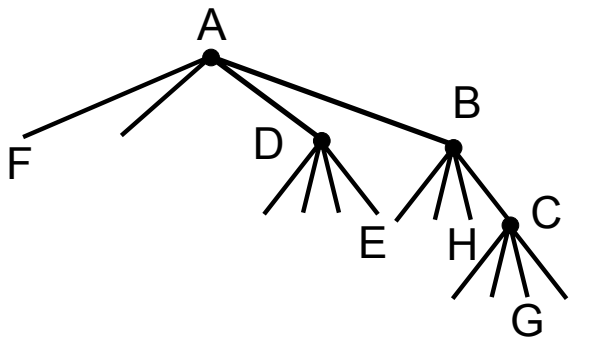
Příklad P kvadrantového stromu





Příklad P kvadrantového stromu





Vložení elementu do P kvadrantového stromu

Postup pro vložení prvku,řekněme $(coord, elem)$, je velmi podobné pro vložení do BVS jen ve dvou rozměrech.

1. Nejprve se podle souřadnice nalezne list, do kterého nově vkládáný element patří
   - Klíči jsou zadané souřadnice (dvě pro 2D prostor), které porovnáváme se středem oblasti, kterou reprezentuje prohledávaný uzel a vybíráme příslušný kvadrant, do kterého souřadnice patří.
2. Pokud je list prázdný (neobsahuje žádný element), pak je element $elem$ vložen do listu.
3. Pokud je list plný, provede se dekompozice uzlu dle obsaženého uzlu
   - Vytvoří se čtyři potomci dekomponovaného uzlu (Středem dekompozice je souřadnice elementu obsaženého v uzlu).
   - Rekursivně se provede operaci vkládání pro dekomponovaný uzel.

Vložení elementu do P kvadrantového stromu

Optimalizované vkládání

1. Souřadnice vkládaných elementů jsou seřazeny podle souřadnice x.
2. Je vybrán element, jehož x-ová souřadnice je mediánem a element je vložen do kořene.
3. Dle vybrané souřadnice jsou elementy rozděleny do čtyř skupin.
4. Pro každou skupinu se rekursivně opakuje tato operace.

Hledání elementu v P kvadrantovém stromu

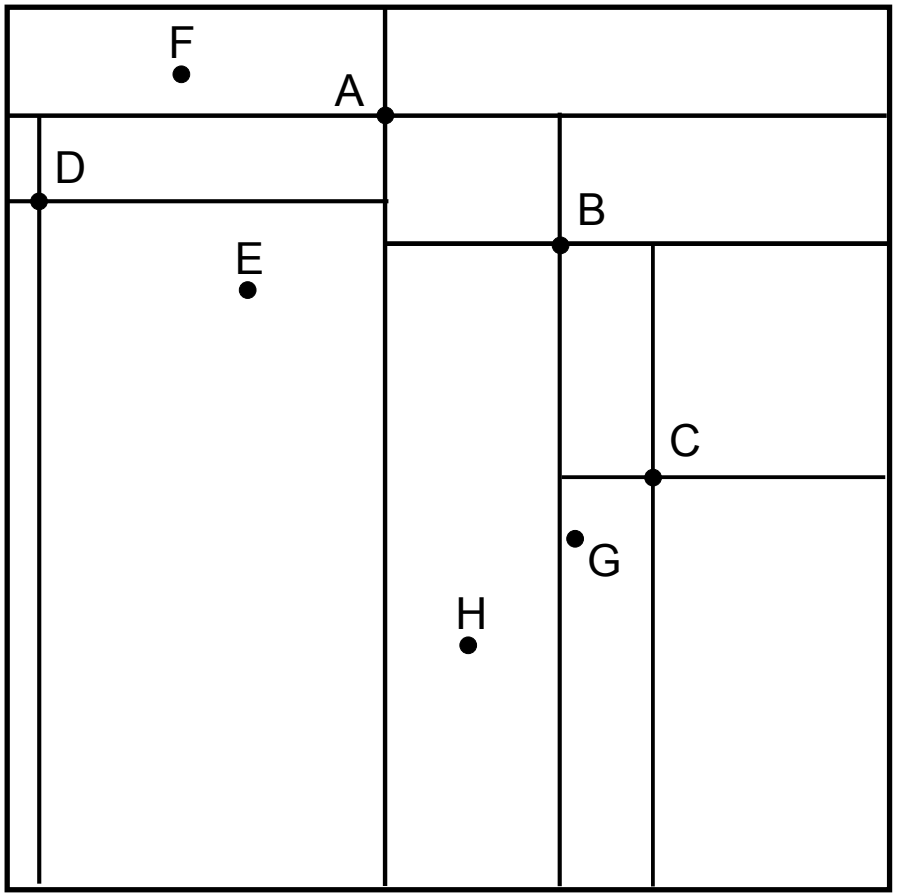
Algoritmus vyhledání oblasti obsahující hledaný element s 2D souřadnicí $coord$ v kvadrantovém stromu s kořenem r:

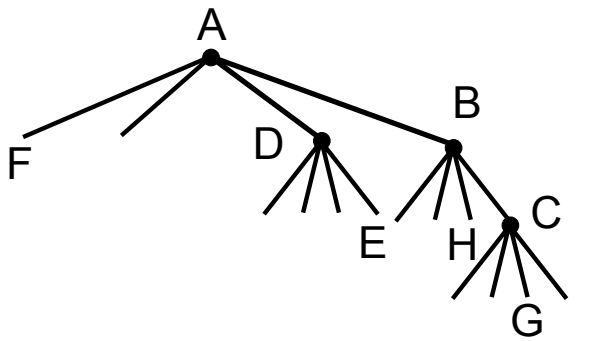
- Otestuje se, zda je kořen r listem:
  - Pokud ano, otestuje se, zda je kořen plný (obsahuje odkaz na uložený element):
    - ★ Pokud ano, provede se operace nad hledaným elementem a zadanou souřadnicí $coord$ (např. detekce kolize souřadnice s uloženým elementem).
    - ★ Pokud ne, je hledání ukončeno jako neúspěšné.
  - Pokud ne, provede se operace porovnání zadané souřadnice se souřadnicí obsaženého elementu.
    - ★ Pokud je shoda souřadnic, element je nalezen.
    - ★ Pokud ne, provede se rekursivně totéž pro odpovídající podstrom (dle hledané souřadnice $coord$) reprezentující jeden z kvadrantů.

Mazání elementu z P kvadrantového stromu

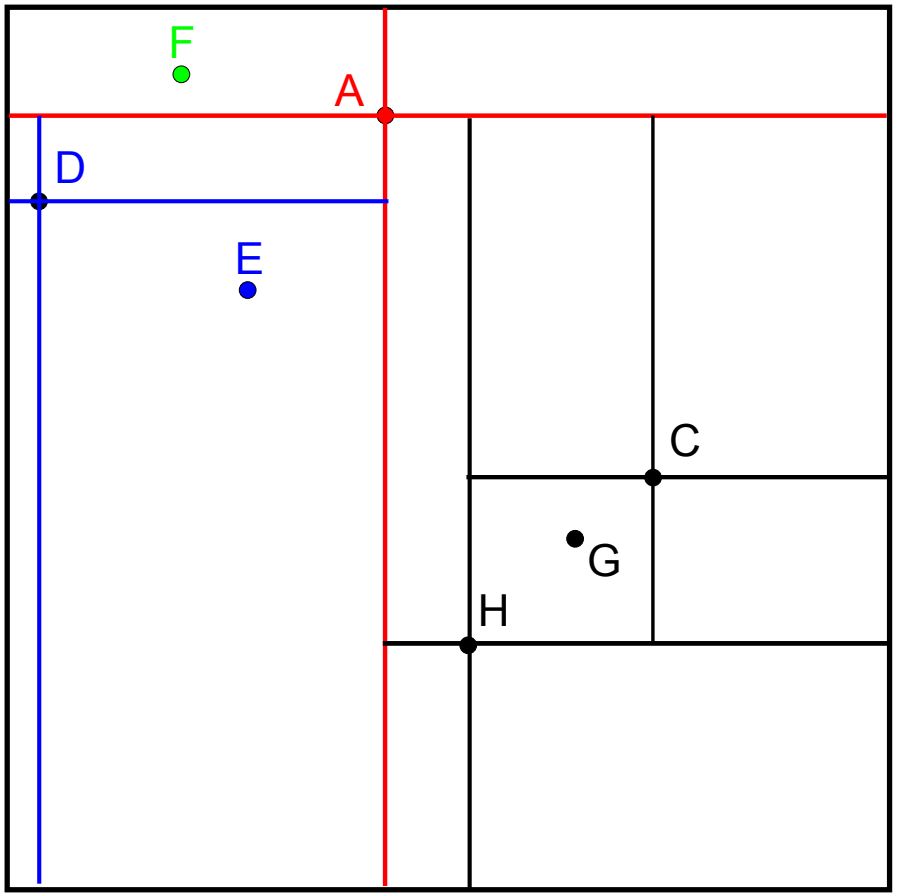
1. Nalezne se uzel, který obsahuje hledaný element.
2. Otestuje se, zda je uzel listem:
   - Pokud ano, odstraní se element a pokud jsou sourozence listy a jsou prázdné, provede se sloučení uzlu (jako u PR).
   - Pokud ne, je element nahrazen elementem ze syna, jehož podstrom má nejmenší hloubku. A rekursivně se provede mazání elementu v synovi, jehož element byl použit pro náhradu.

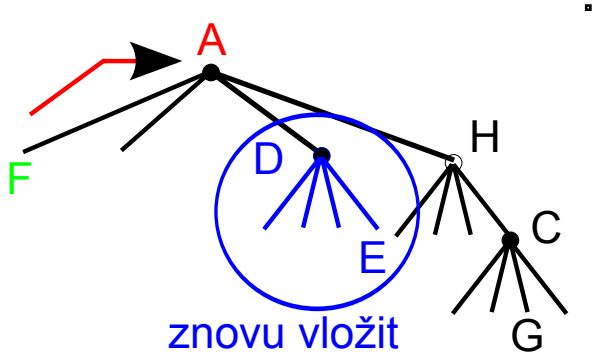
Mazání v P kvadrantového stromu



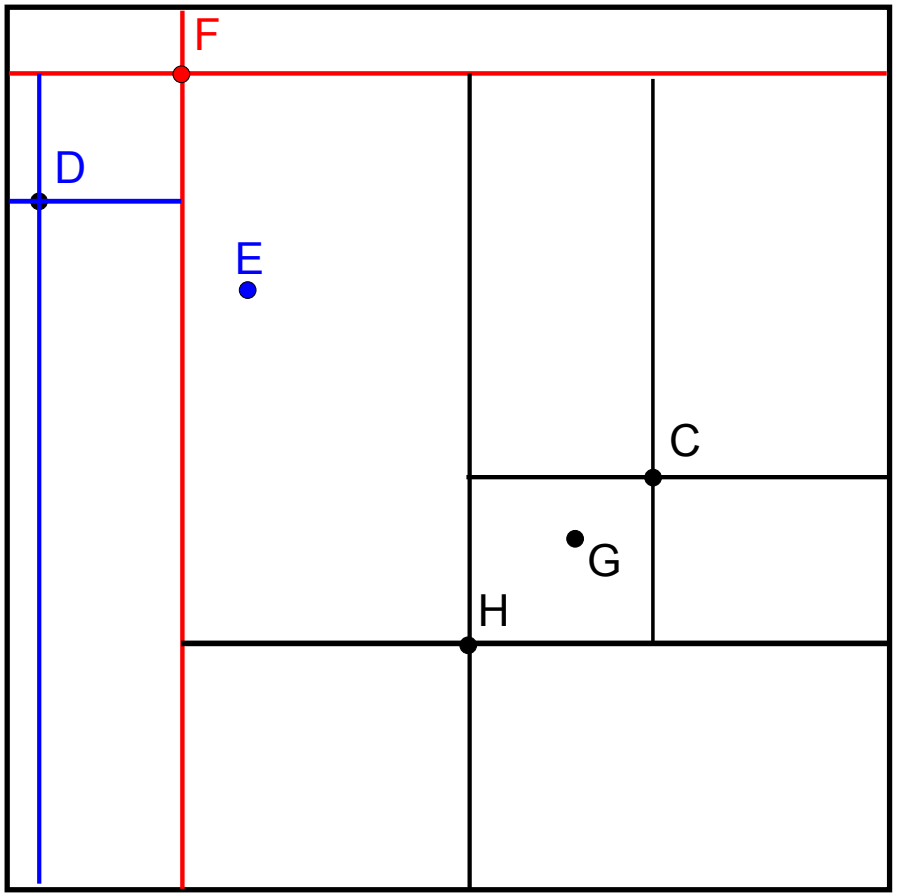


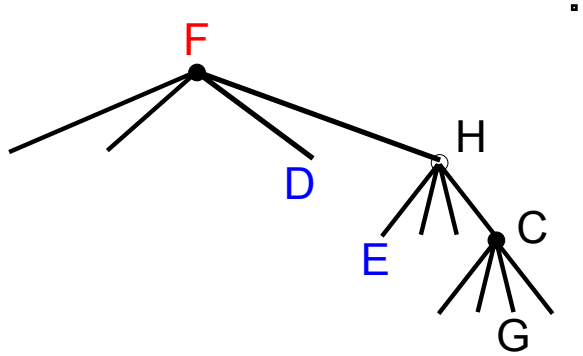
Mazání v P kvadrantového stromu



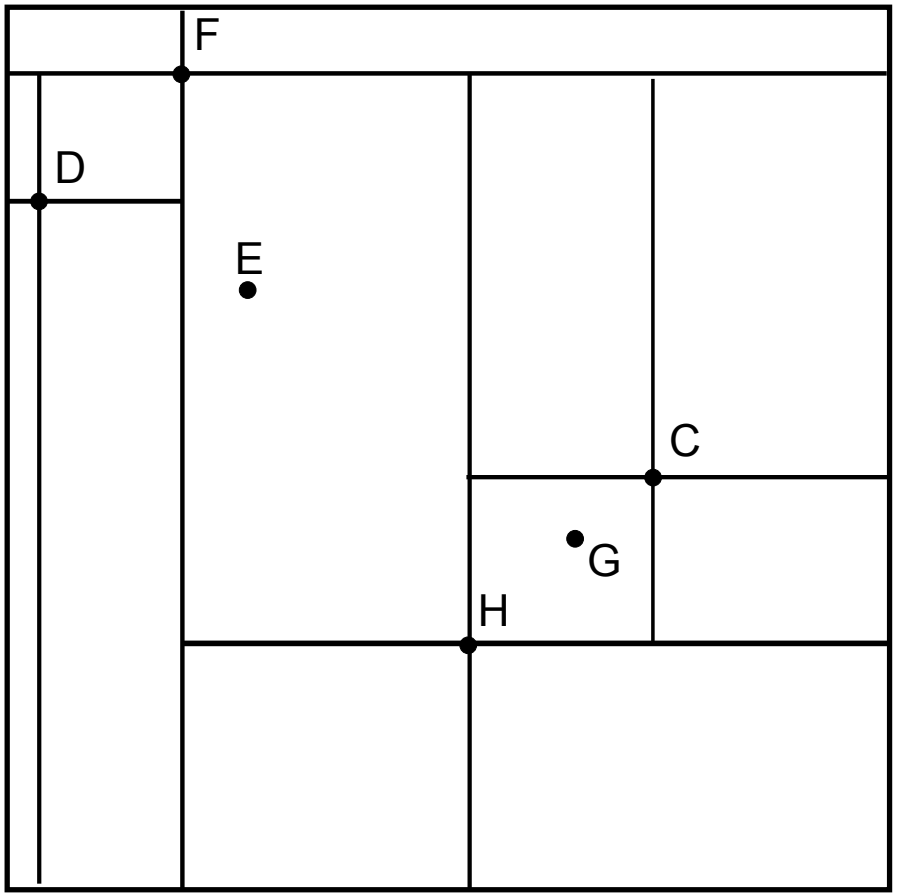


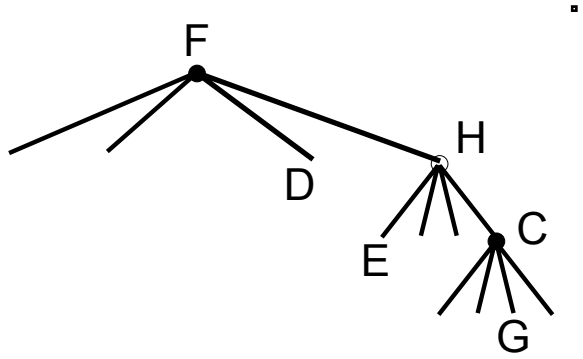
Mazání v P kvadrantového stromu





Mazání v P kvadrantového stromu



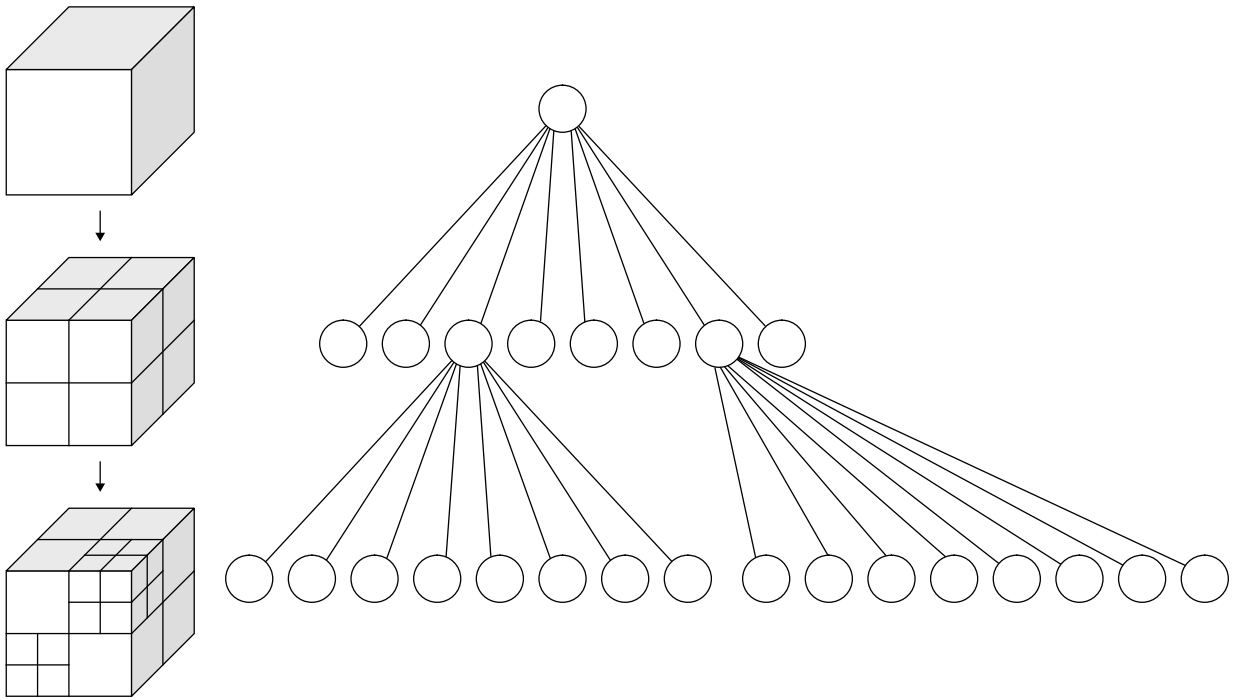


Hodnocení P kvadrantového stromu

1. Struktura PR kvadrantového stromu je závislá na pořadí vkládaných elementů.
2. Strom lze konstruovat vyvážený (operace mají $log_4 N$ složitost).
3. Počet uzlů stromu je blízký počtu vložených elementů

Oktalový strom

- Obdoba kvadrantového stromu v třírozměrném prostoru.
- Počet potomků vnitřního uzlu je právě 8.
- Používá se pro hierarchickou reprezentaci 3D scén.



K-D strom

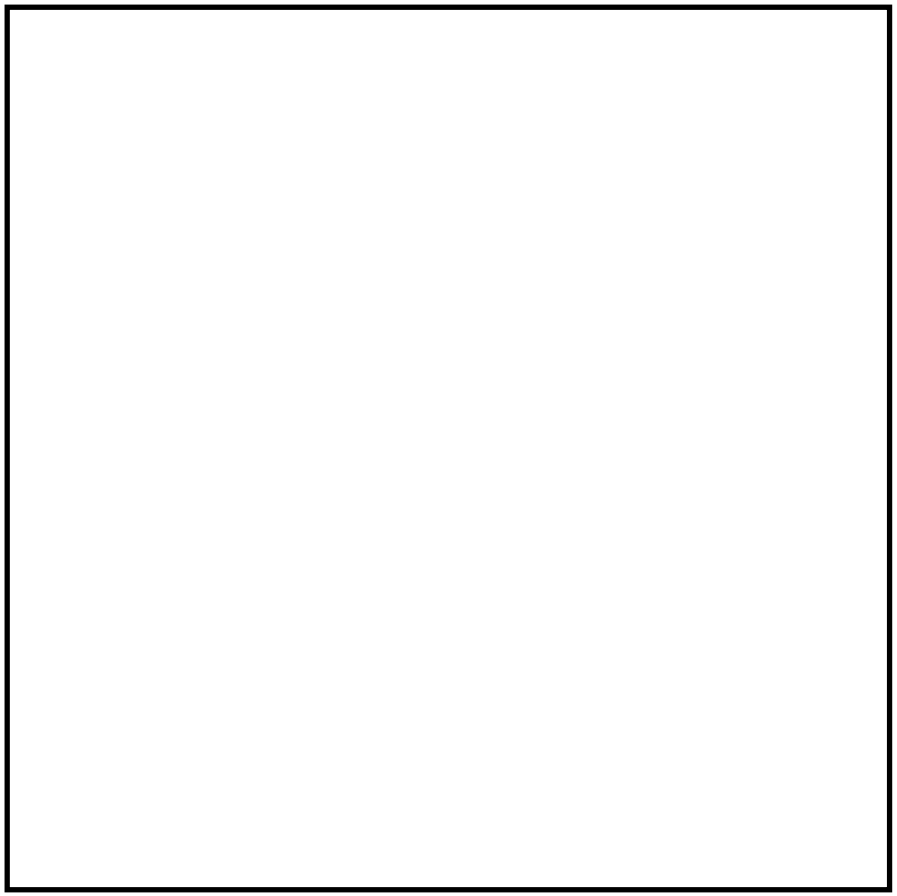
- Autor J.L. Bentley,

Definice (K-D strom)

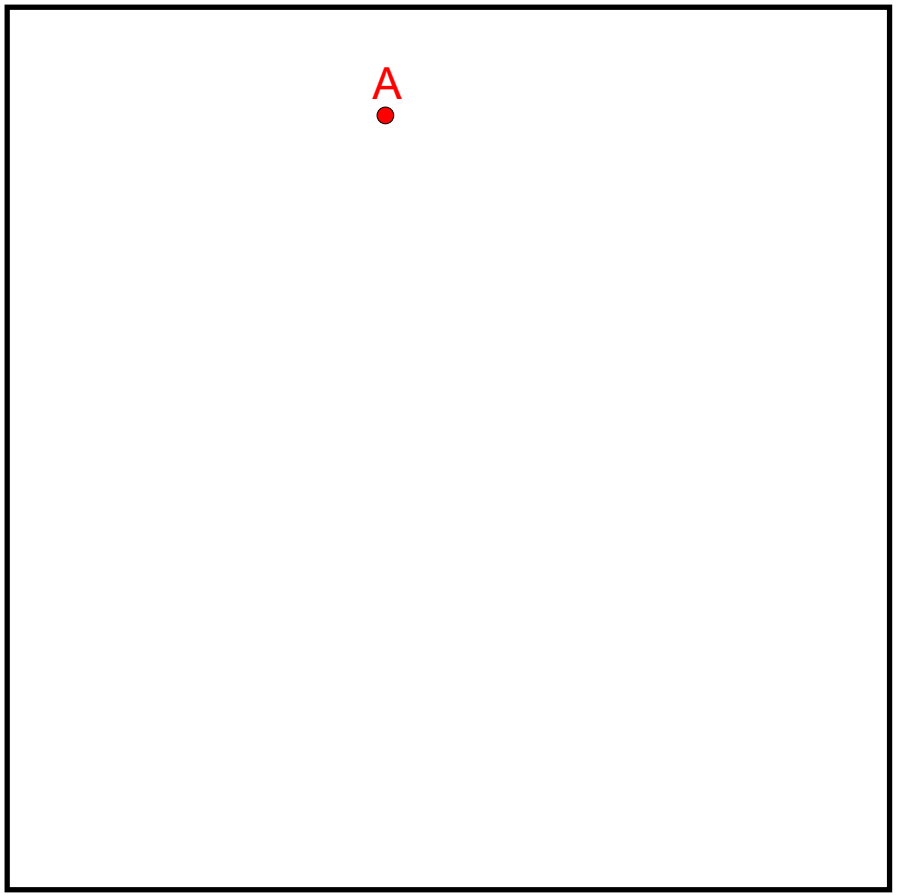
K-D strom je BVS provádějící dělení střídavě podle x-ové a y-ové souřadnice

1. každý vnitřní uzel reprezentuje jednu dekompozici 2D prostoru na dvě navzájem disjunktní oblasti,
2. existuje několik variant K-D Stromu:
   - dekomponované oblasti jsou vždy stejně veliké (PR K-D strom),
   - listy jsou vždy prázdné,
   - dekompozice je prováděna dle jedné souřadnice vloženého elementu (tuto variantu budeme prezentovat)
3. každý jeho vnitřní uzel má právě 2 potomky,
4. každý uzel obsahuje navíc d-rozměrný klíč, podle kterého je prováděna dekompozice,

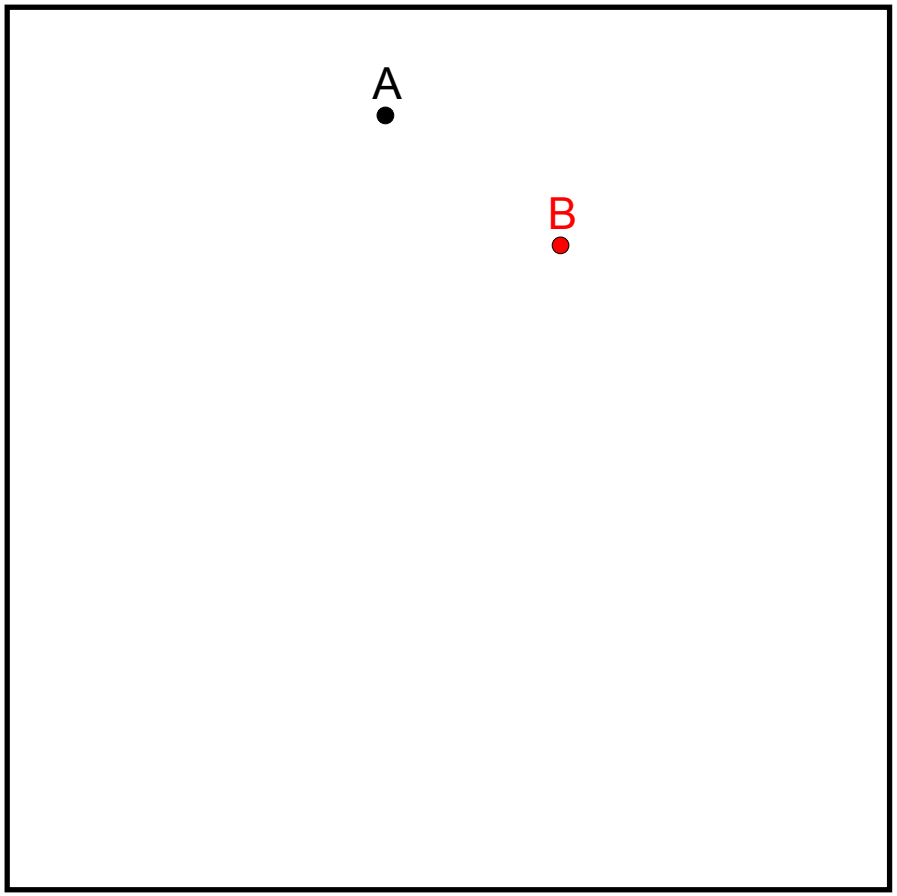
Příklad 2D K-D stromu



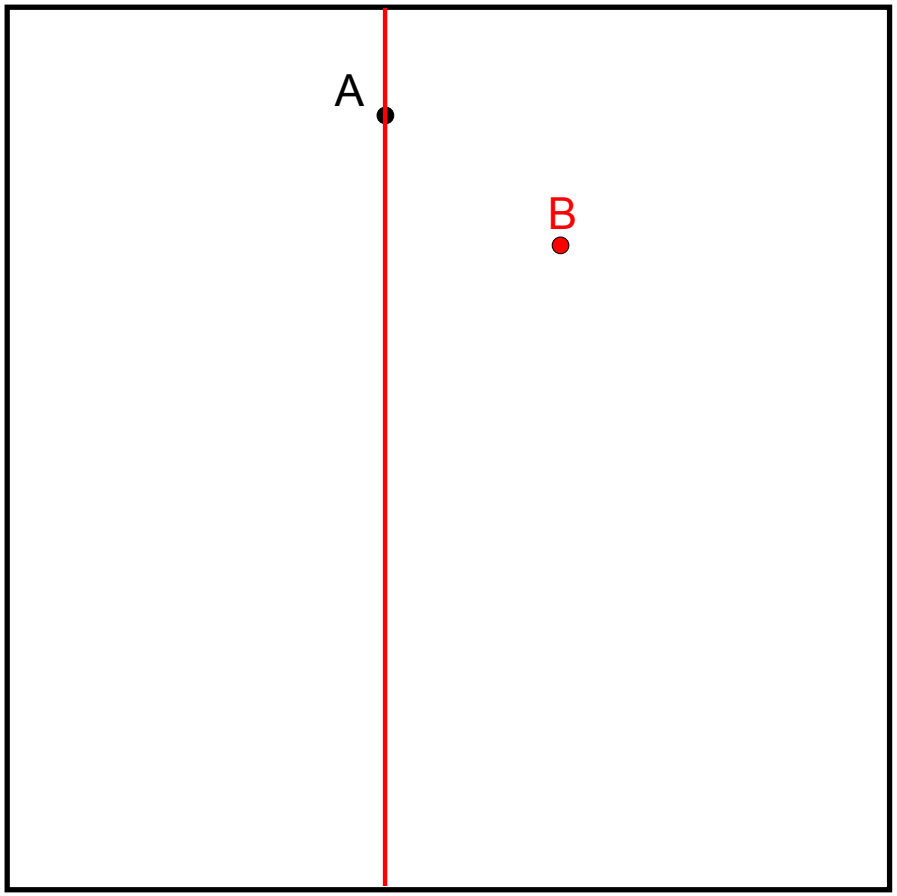
Příklad 2D K-D stromu

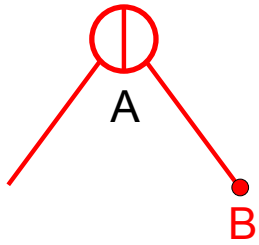


Příklad 2D K-D stromu

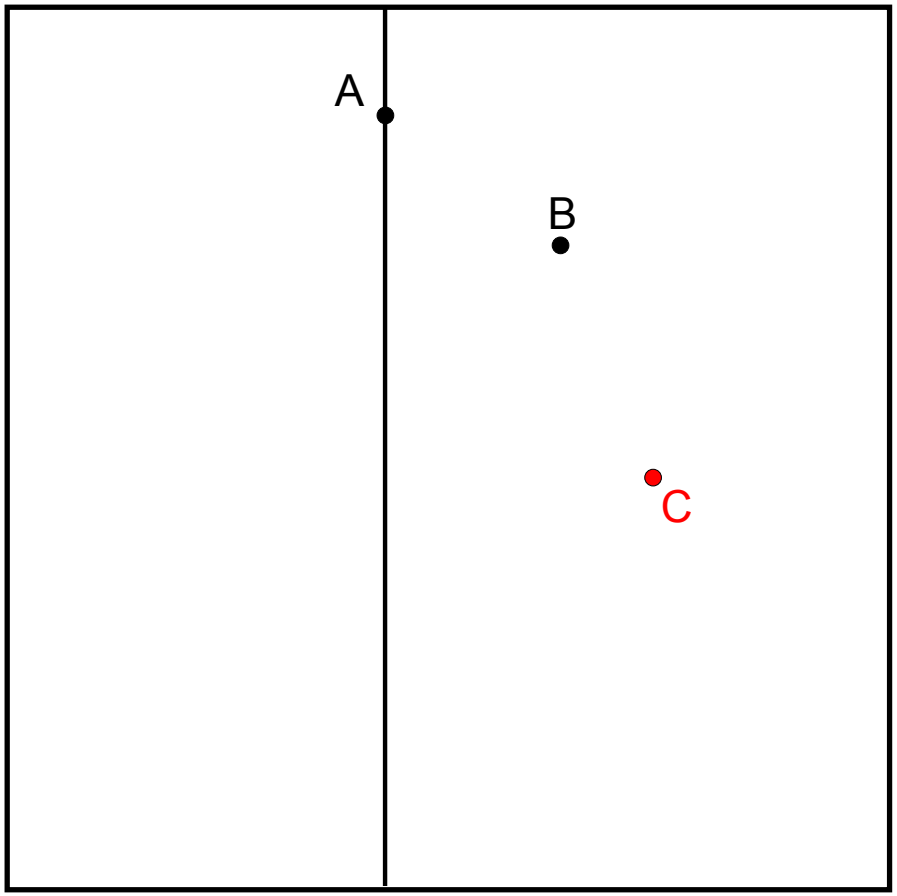


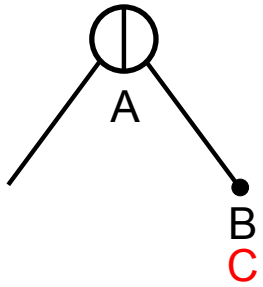
Příklad 2D K-D stromu



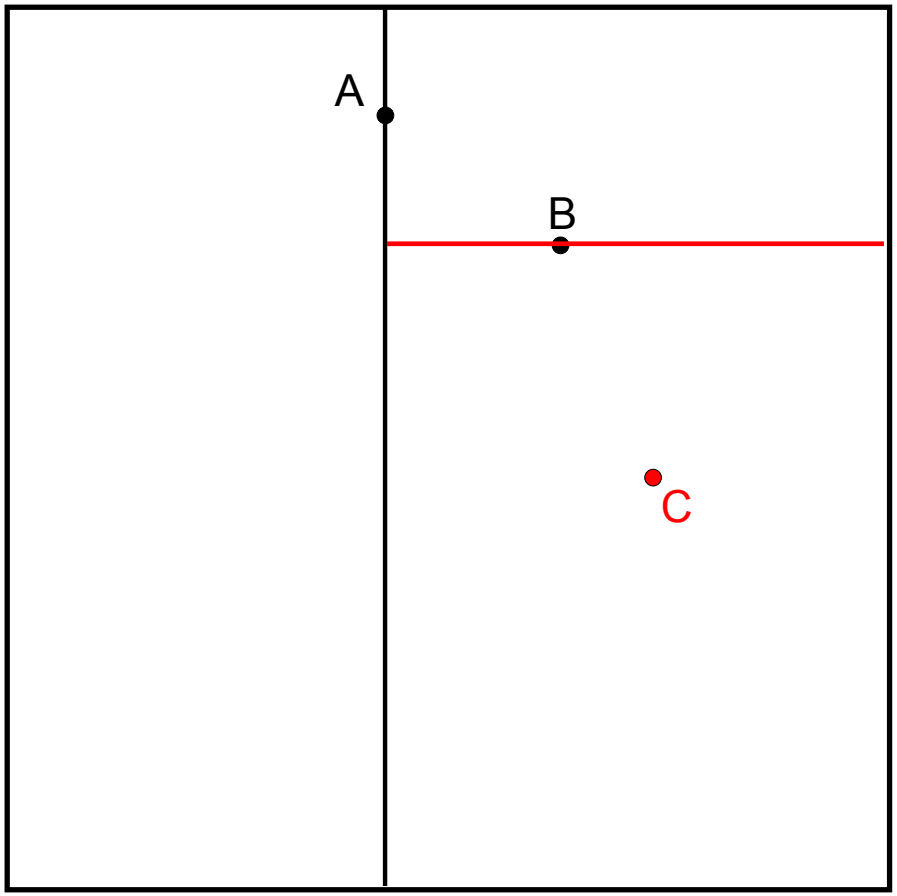


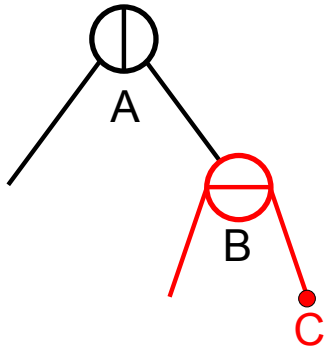
Příklad 2D K-D stromu



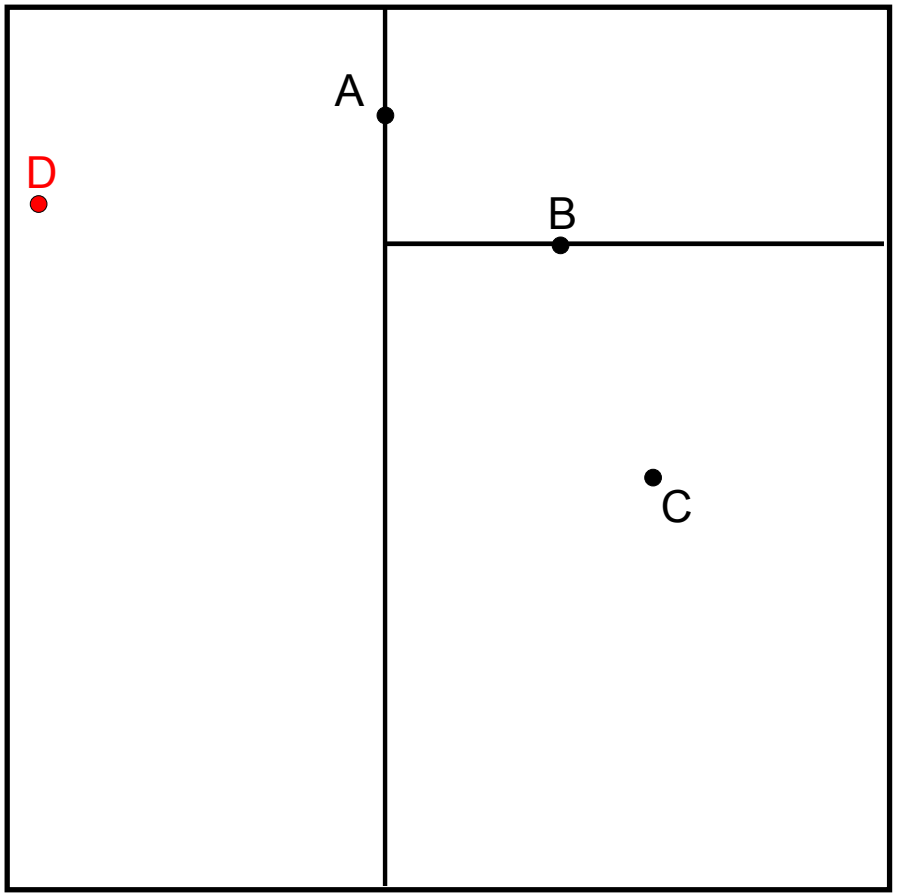


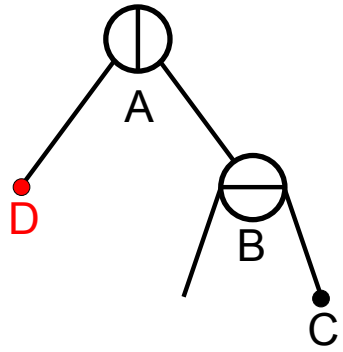
Příklad 2D K-D stromu



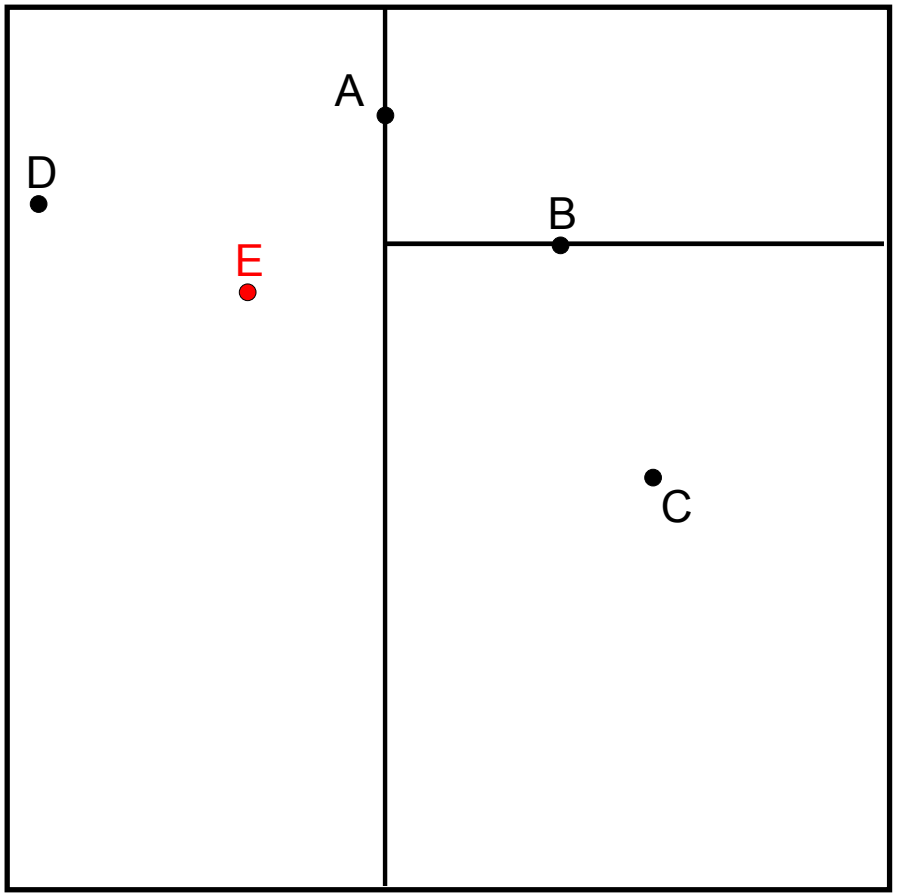


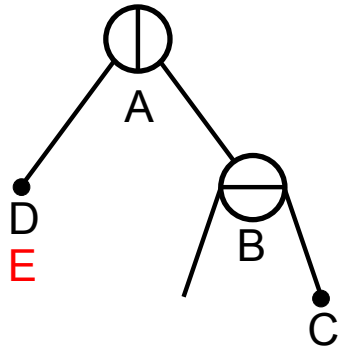
Příklad 2D K-D stromu



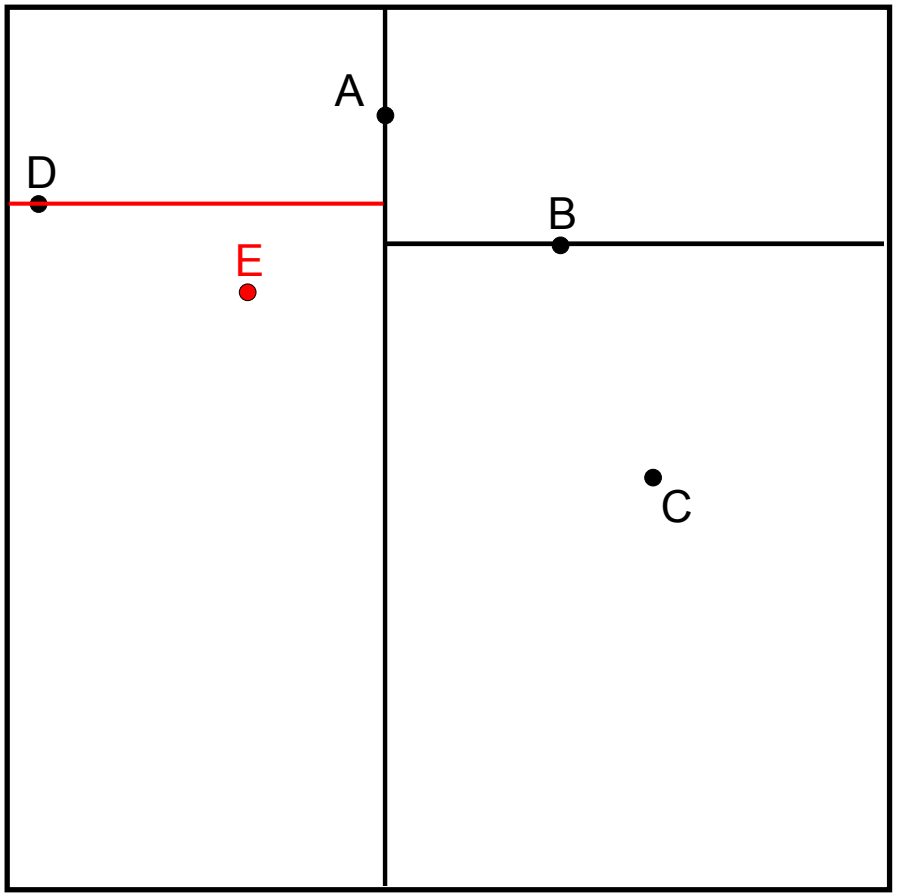


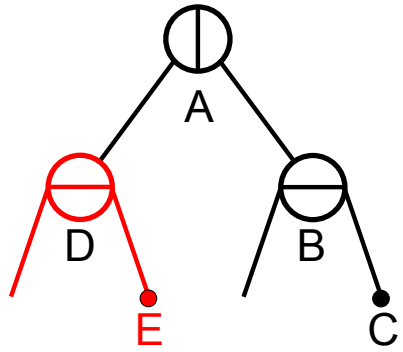
Příklad 2D K-D stromu



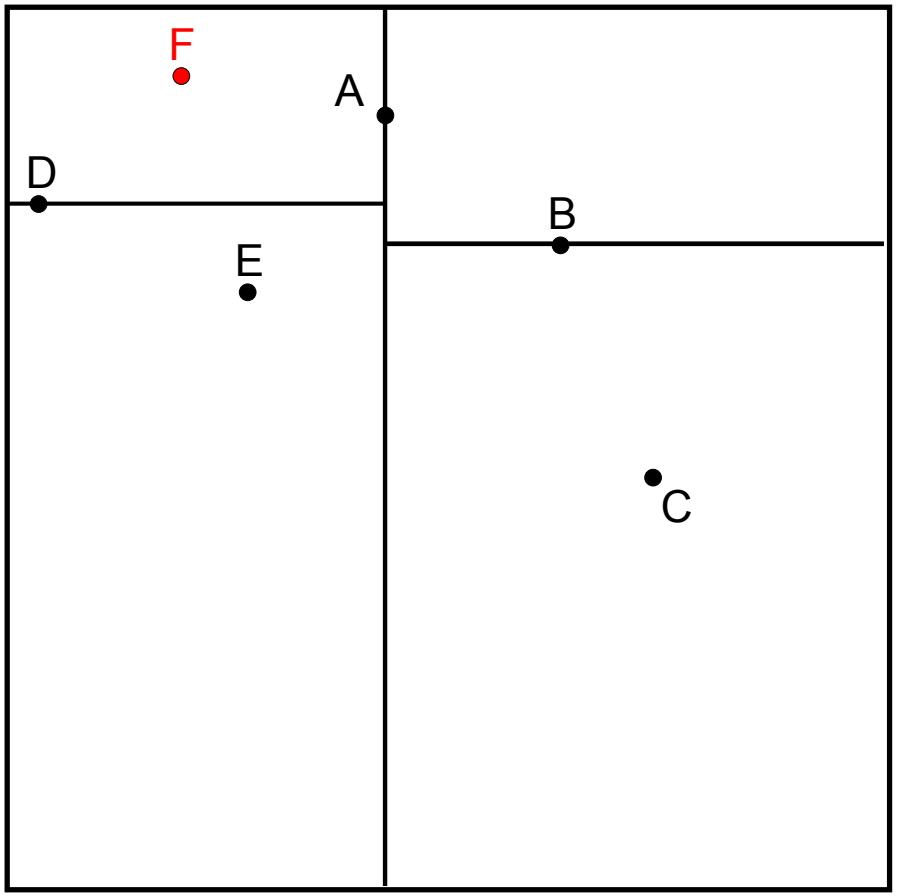


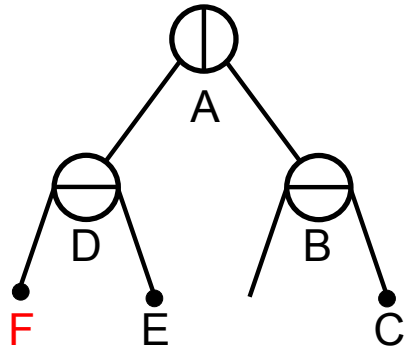
Příklad 2D K-D stromu



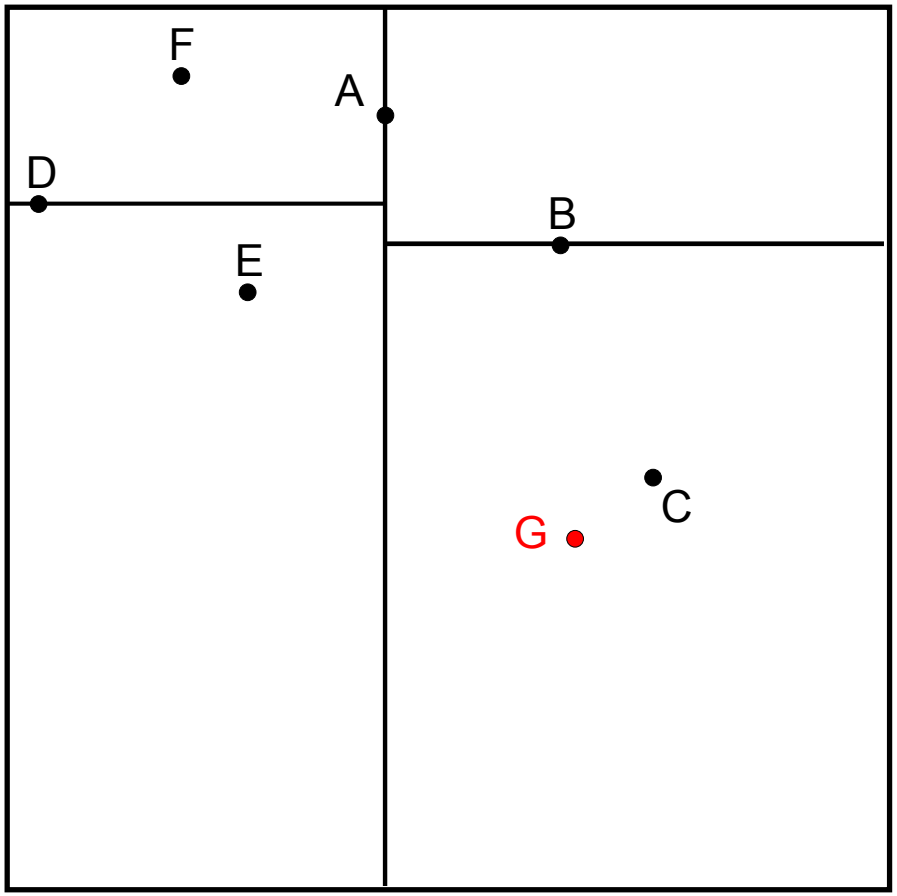


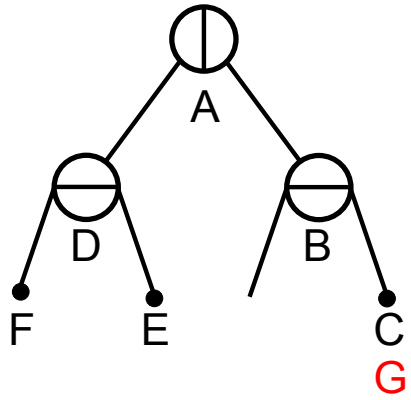
Příklad 2D K-D stromu



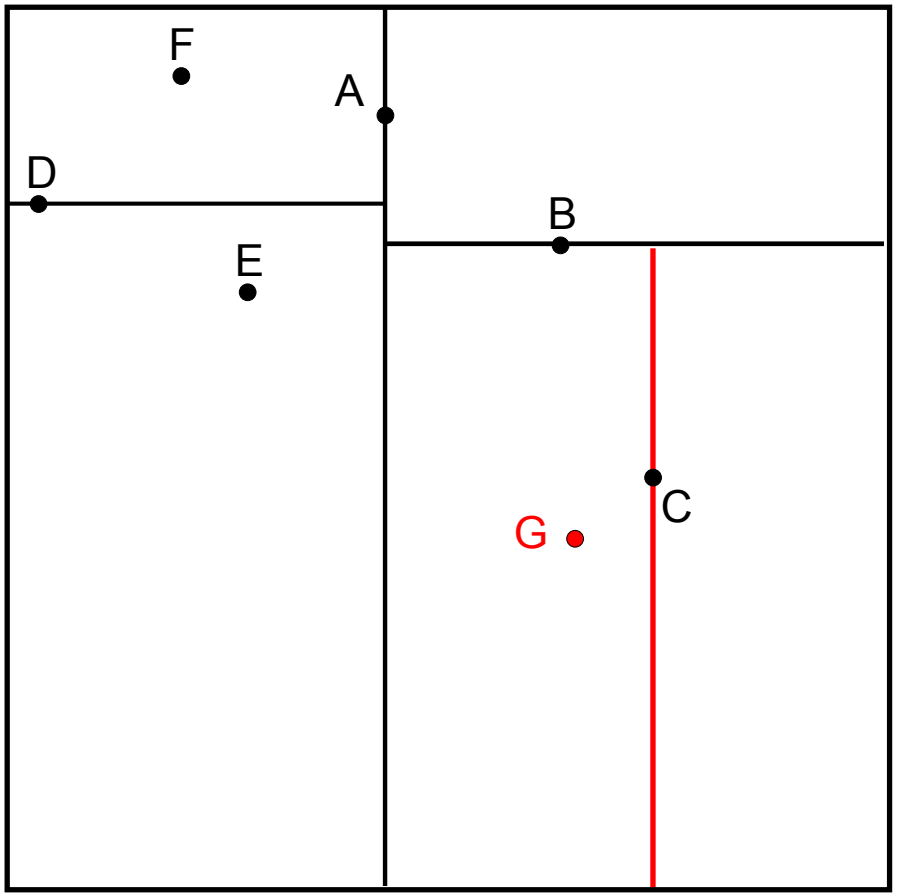


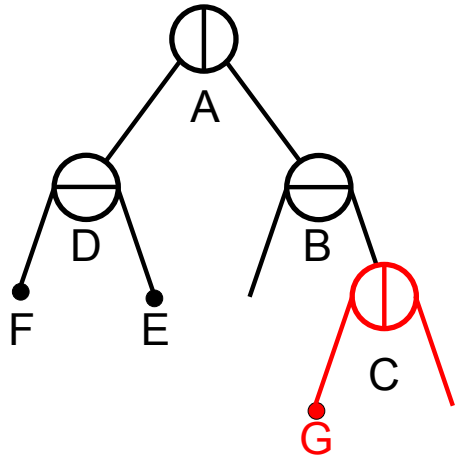
Příklad 2D K-D stromu



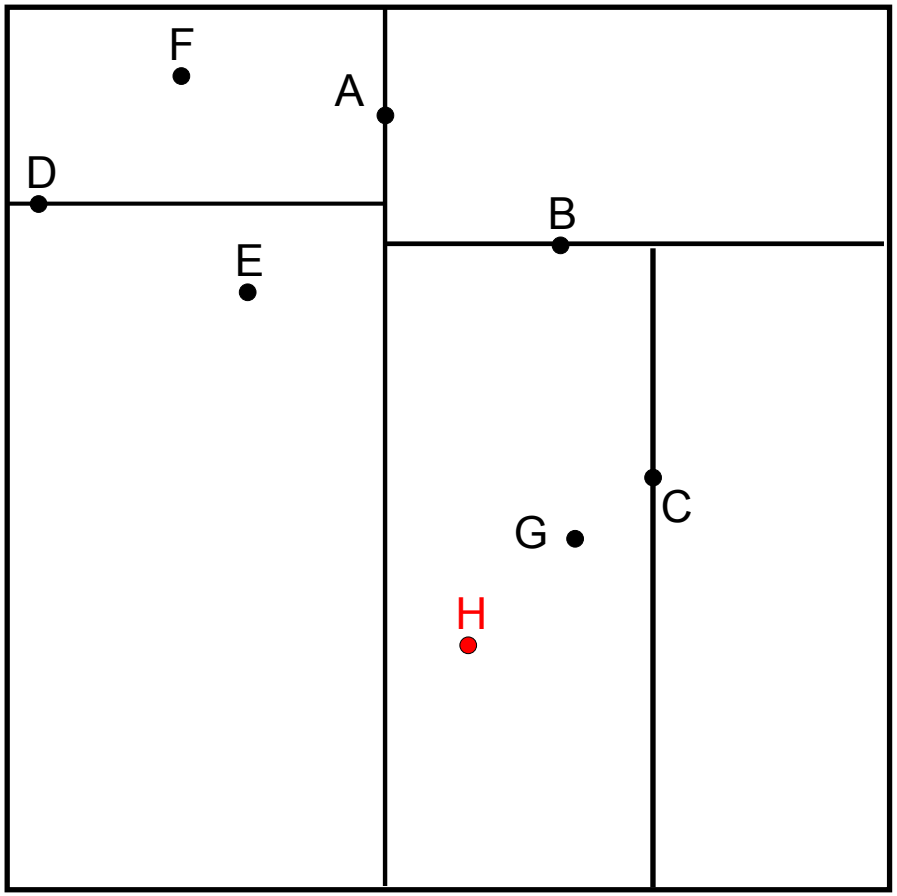


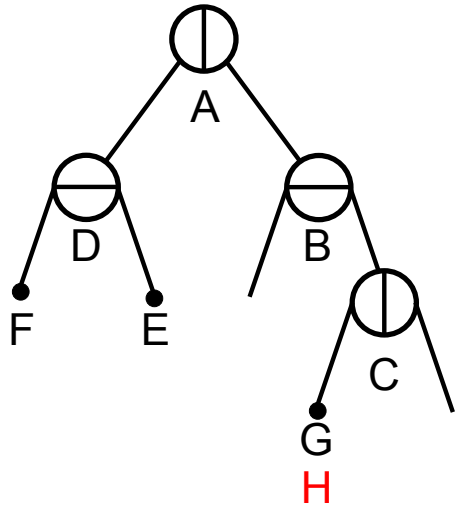
Příklad 2D K-D stromu



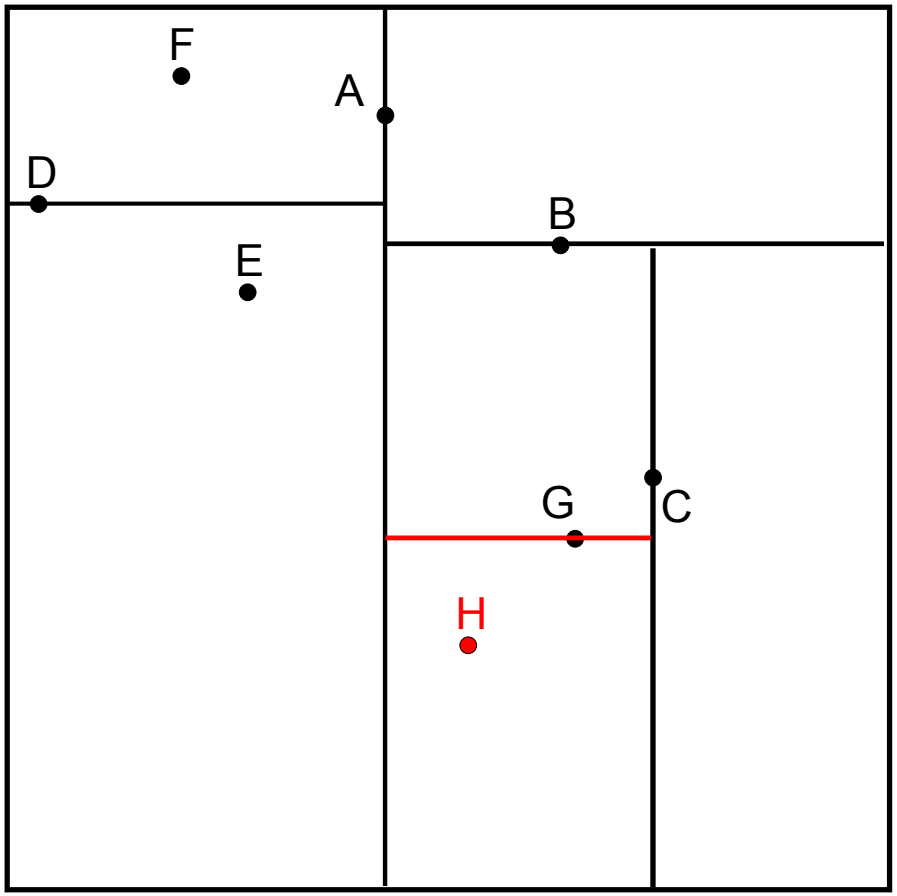


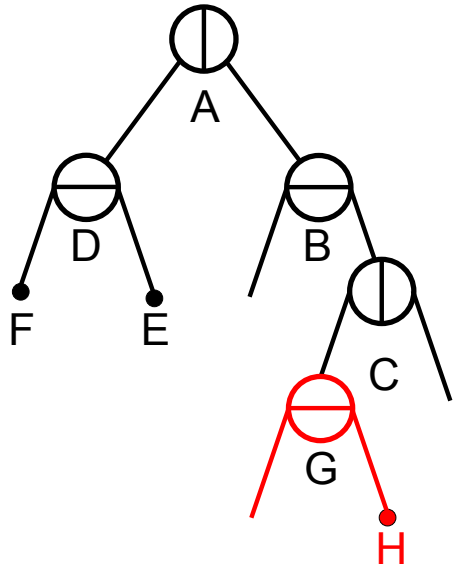
Příklad 2D K-D stromu



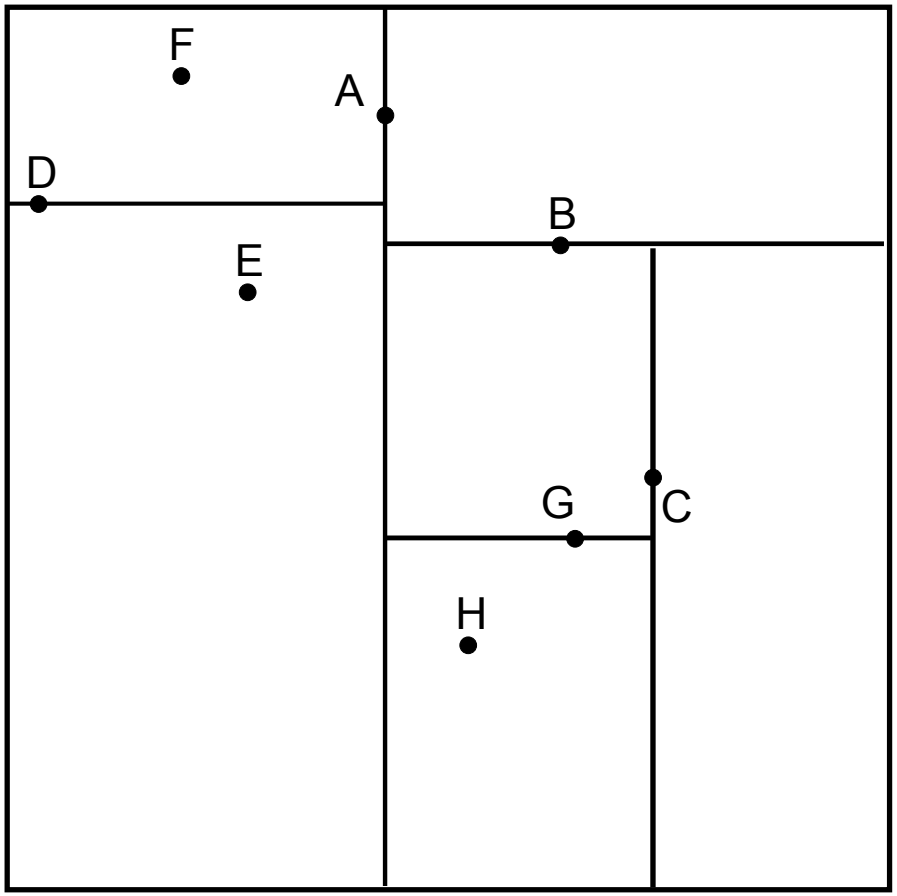


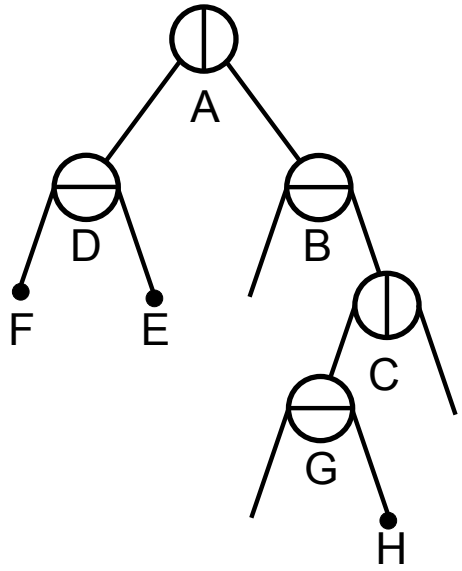
Příklad 2D K-D stromu





Příklad 2D K-D stromu





Vložení elementů do K-D stromu

- Postup pro vložení elementu, je podobné jako vložení do BVS. Rozdílem je, že pro porovnávání cyklicky používáme jednotlivé dimense prvků.
- Takový postup může způsobit, že K-D strom bude nevyvážený.
- Pro vyvažování nelze použít rotace, protože strom je uspořádán podle více dimensí.
- Vyvažovaní je poměrně obtížná záležitost, kterou řeší K-D-B stromy, hB stromy a Bkd stromy.

Vložení elementu do K-D stromu

Postup pro vložení prvku,řekněme $(coord, elem)$, je velmi podobné pro vložení do BVS.

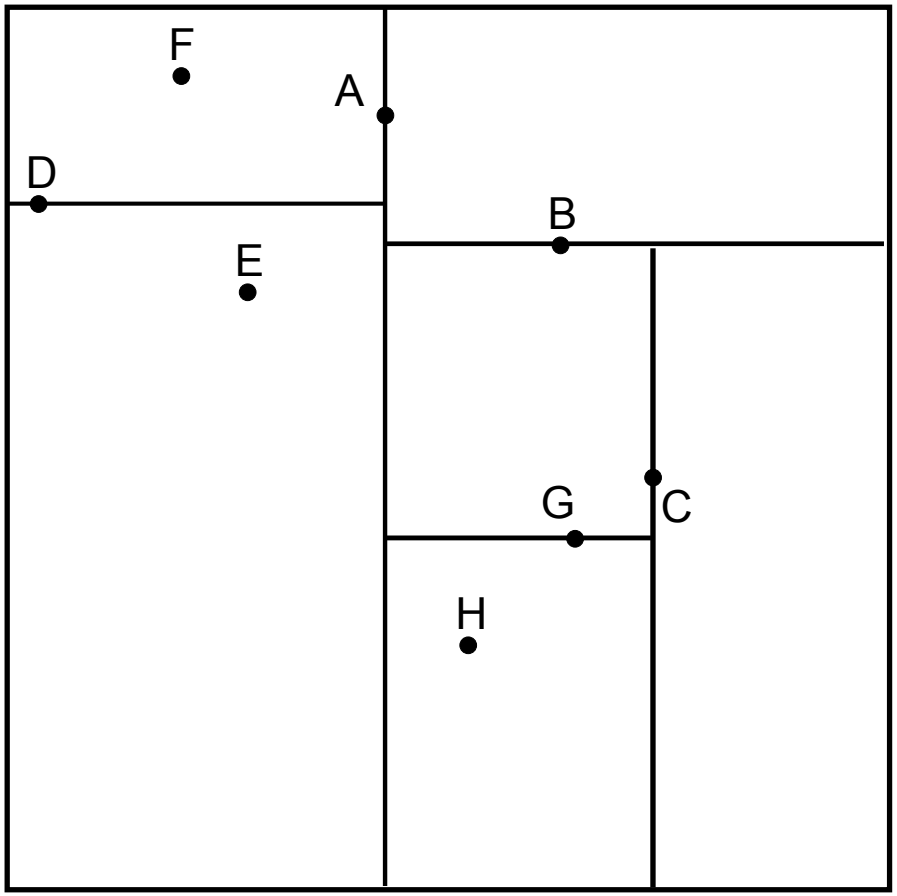
1. Nejprve se podle souřadnice nalezne list, do kterého nově vkládaný element patří
   - Klíčem je k-tá souřadnice (k = hloubka mod dimense), která reprezentuje rozdělení oblasti podle k-té dimense na dvě disjunktní podoblasti.
2. Provede se dekompozice uzlu:
   - Vytvoří se dva potomci dekomponovaného uzlu a nastaví se jeho klíč tak, aby původní a vkládaný element byly v různých oblastech.
   - Element původně uložený v dekomponovaném uzlu se přesune do odpovídajícího syna (dle souřadnice uloženého elementu) a nově vkládaný do druhého.

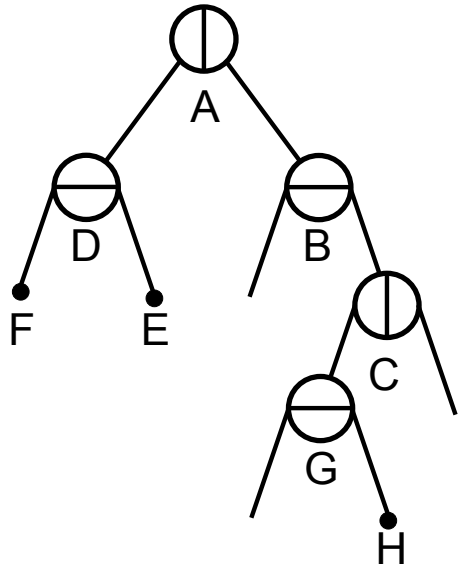
Mazání bodu z K-D stromu

1. Nalezne se uzel, který obsahuje hledaný bod.
2. Pokud existuje:
   1. Vymaže se.
   2. Pokud je uzel listem a jeho sourozenec neobsahuje bod, stane se jeho otec listem.
   3. Pokud není uzel listem, musí být vymazaný bod nahrazen jiným.

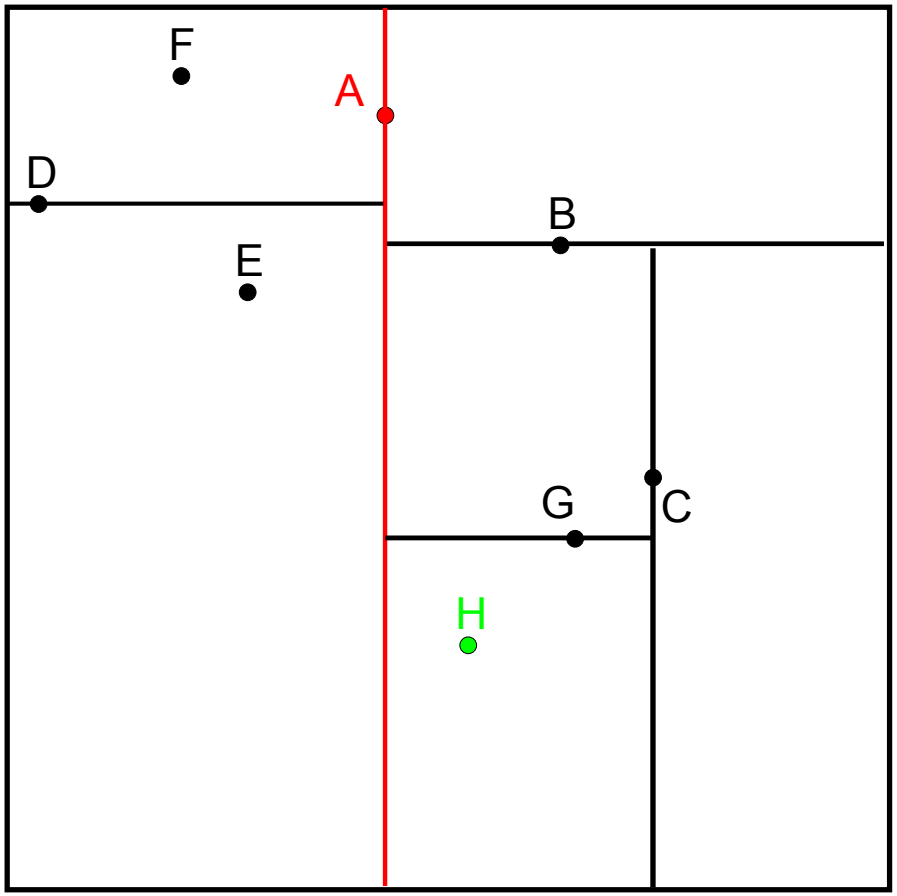
      1. Pokud uzel dělil scénu podle x-ové souřadnice, najdeme uzel R obsahující bod s maximální x-ovou souøadnicí v levém podstromu nebo s minimální x-ovou souřadnicí v pravém podstromu, nalezený bod použijeme jako náhradu a stejný postup uplatníme na uzel R.
      2. Pokud uzel dělil scénu podle y-ové souřadnice, najdeme uzel R obsahující bod s maximální y-ovou souøadnicí v levém podstromu nebo s minimální y-ovou souřadnicí v pravém podstromu, nalezený bod použijeme jako náhradu a stejný postup uplatníme na uzel R.

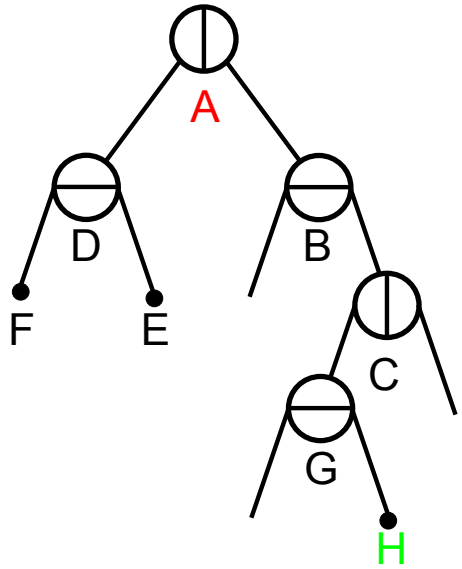
Příklad mazaní z 2D K-D stromu



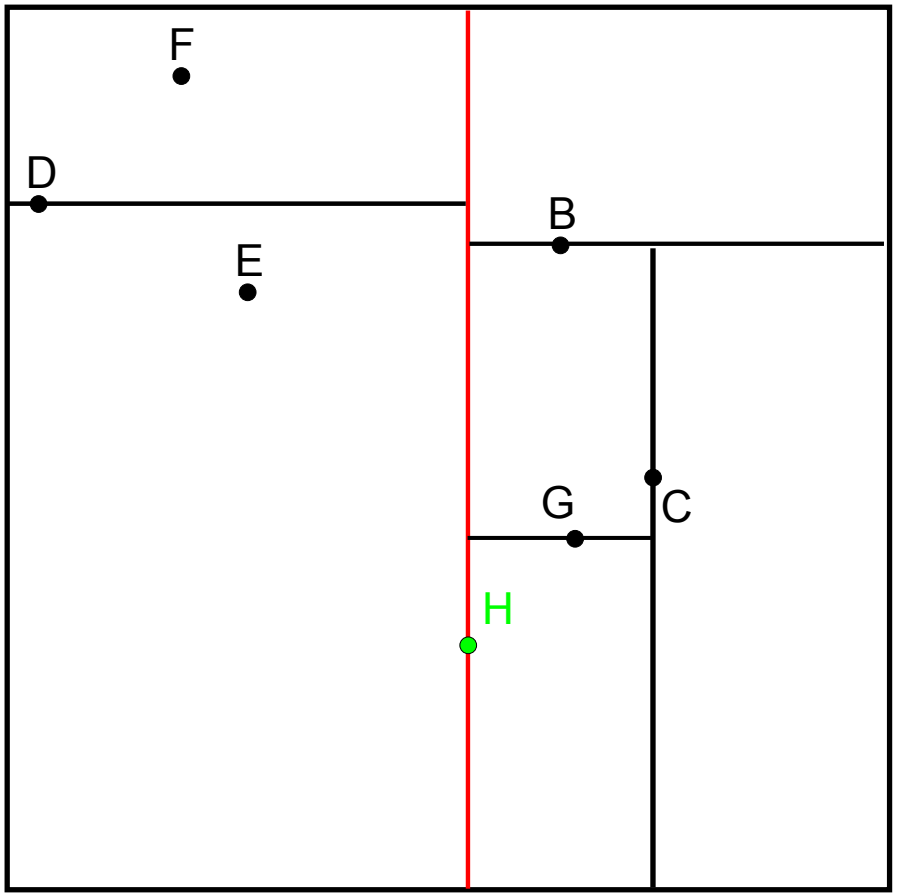


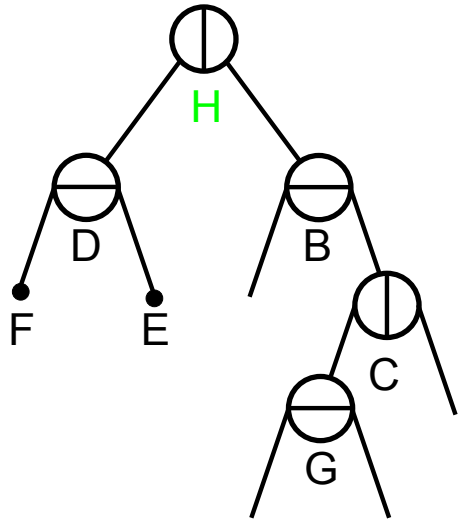
Příklad mazaní z 2D K-D stromu



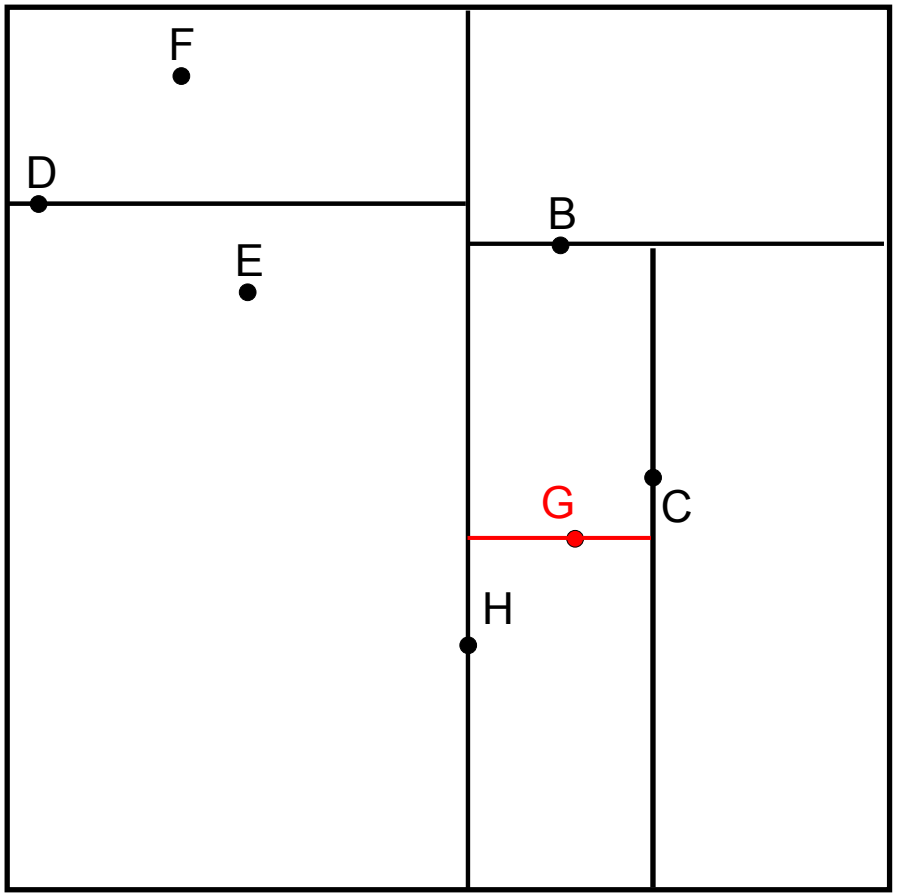


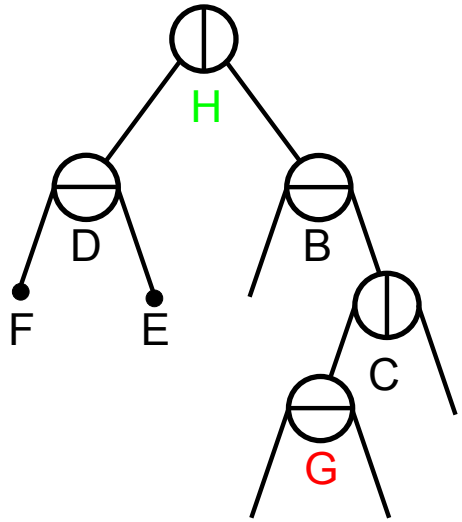
Příklad mazaní z 2D K-D stromu



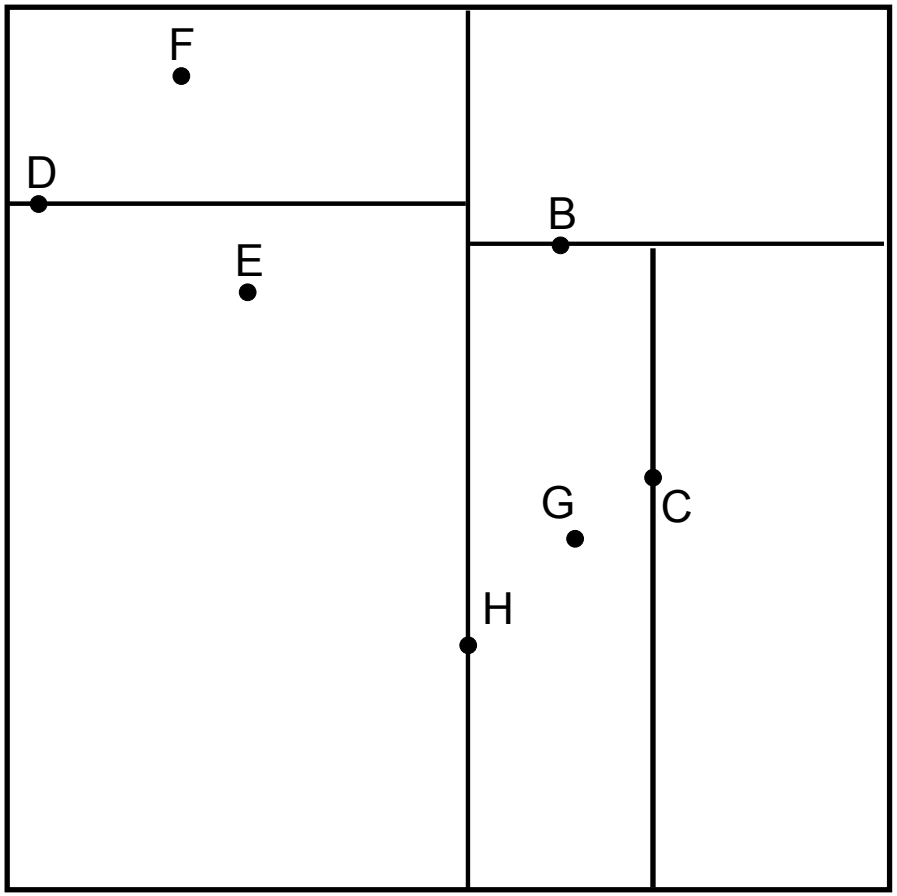


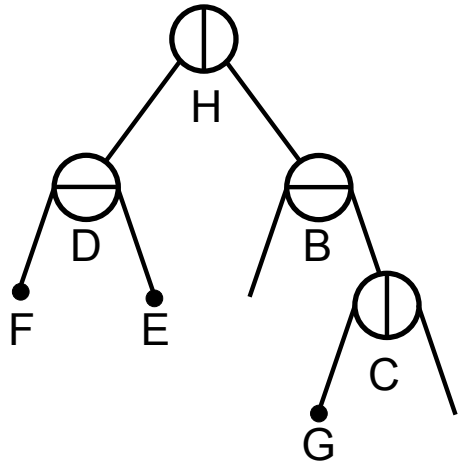
Příklad mazaní z 2D K-D stromu





Příklad mazaní z 2D K-D stromu





Hledání konvexní obálky

Definice

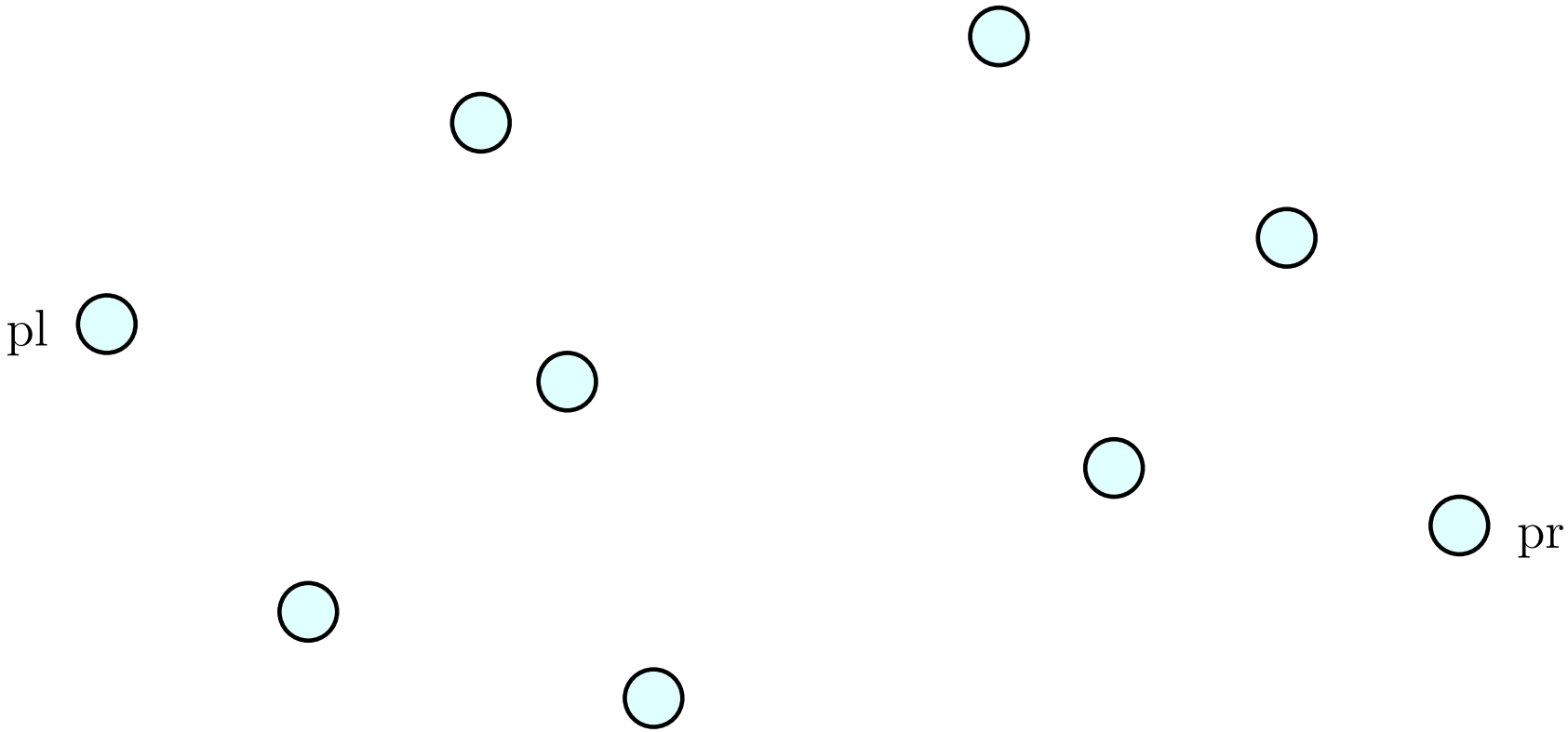
Nechť **V** je množina bodů nebo jiných objektů v rovině. Konvexní obal (convex hull) je nejmenší konvexní objekt, který obsahuje všechny body nebo objekty **V**.

- Vstupem je množina bodů.
- Výstupem je uspořádaný seznam (ve směru hodinových ručiček) bodů,reprezentující konvexní obálku.

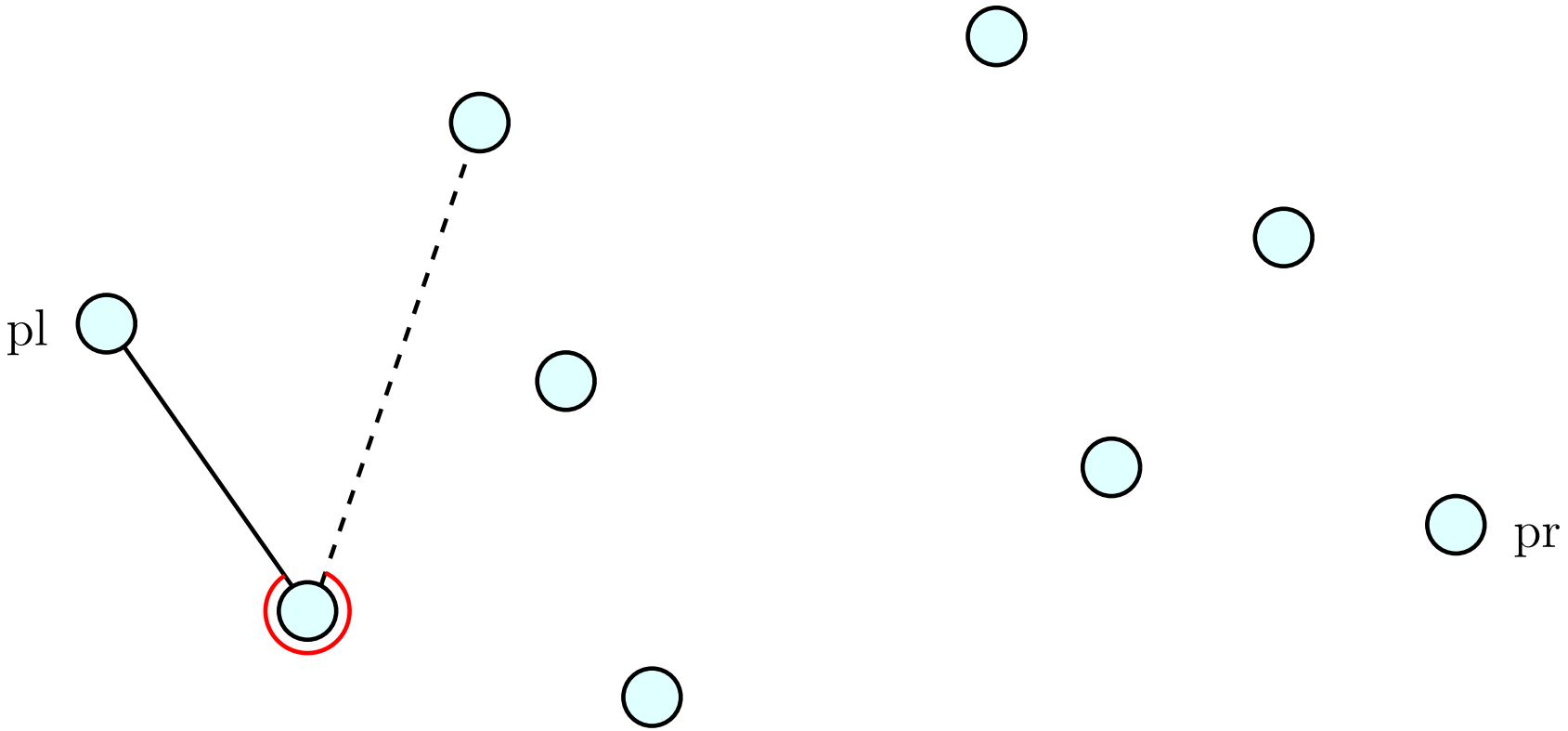
Inkrementální Grahamův algoritmus hledání konvexní obálky

1. Seřadíme množinu bodů dle x-ové souřadnice.
2. Nejprve nalezneme horní část konvexní obálky a to tak, že postupně vkládáme do výstupního pole body ze vstupní množiny.
3. Testujeme, zda poslední tři body tvoří konvexní úhel. Pokud poslední tři body netvoří konvexní úhel, je z výstupní množiny odstraněn předposlední bod a test opakujeme
4. Obdobným způsobem nalezneme dolní část konvexní obálky.

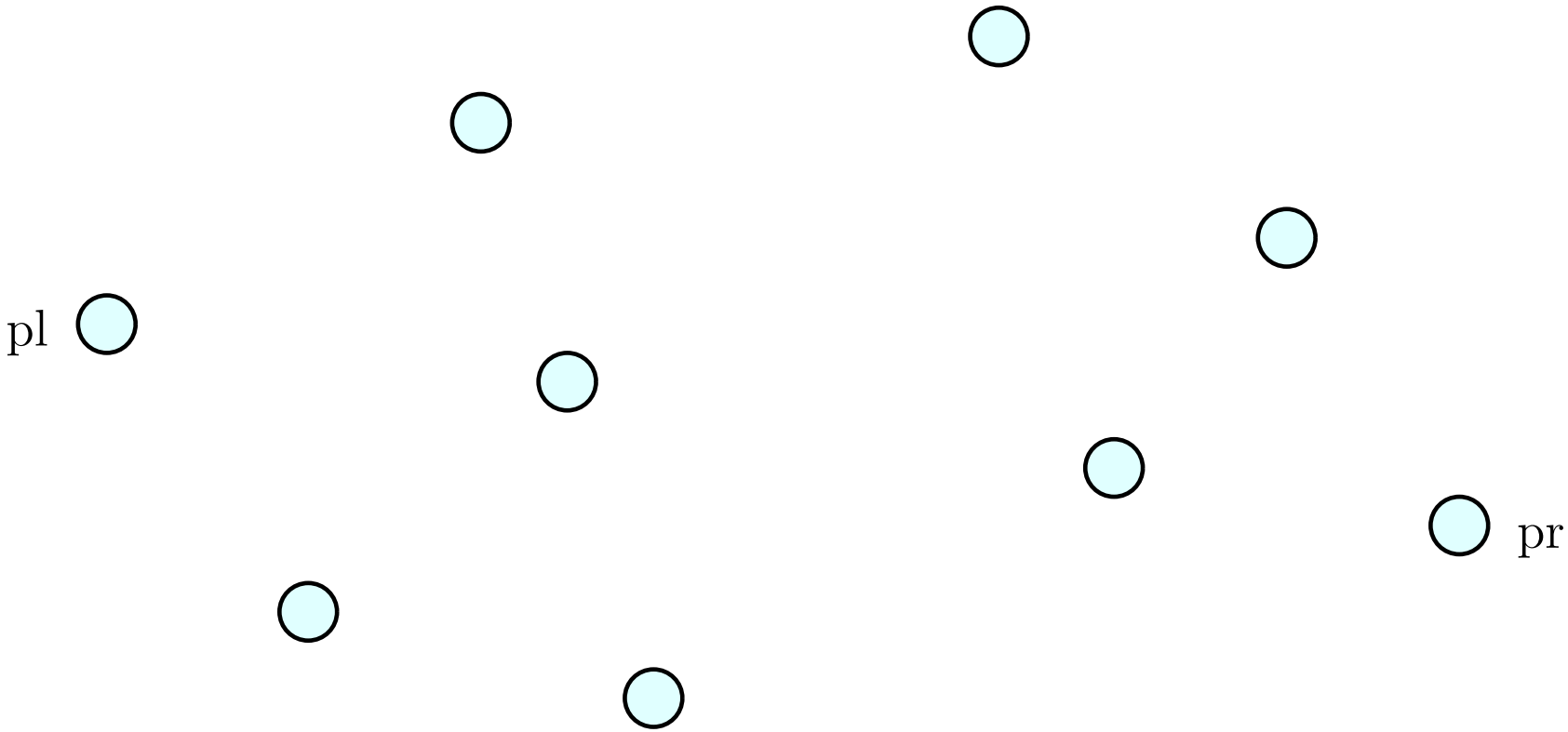
Grahamův algoritmus konstrukce konvexní obálky



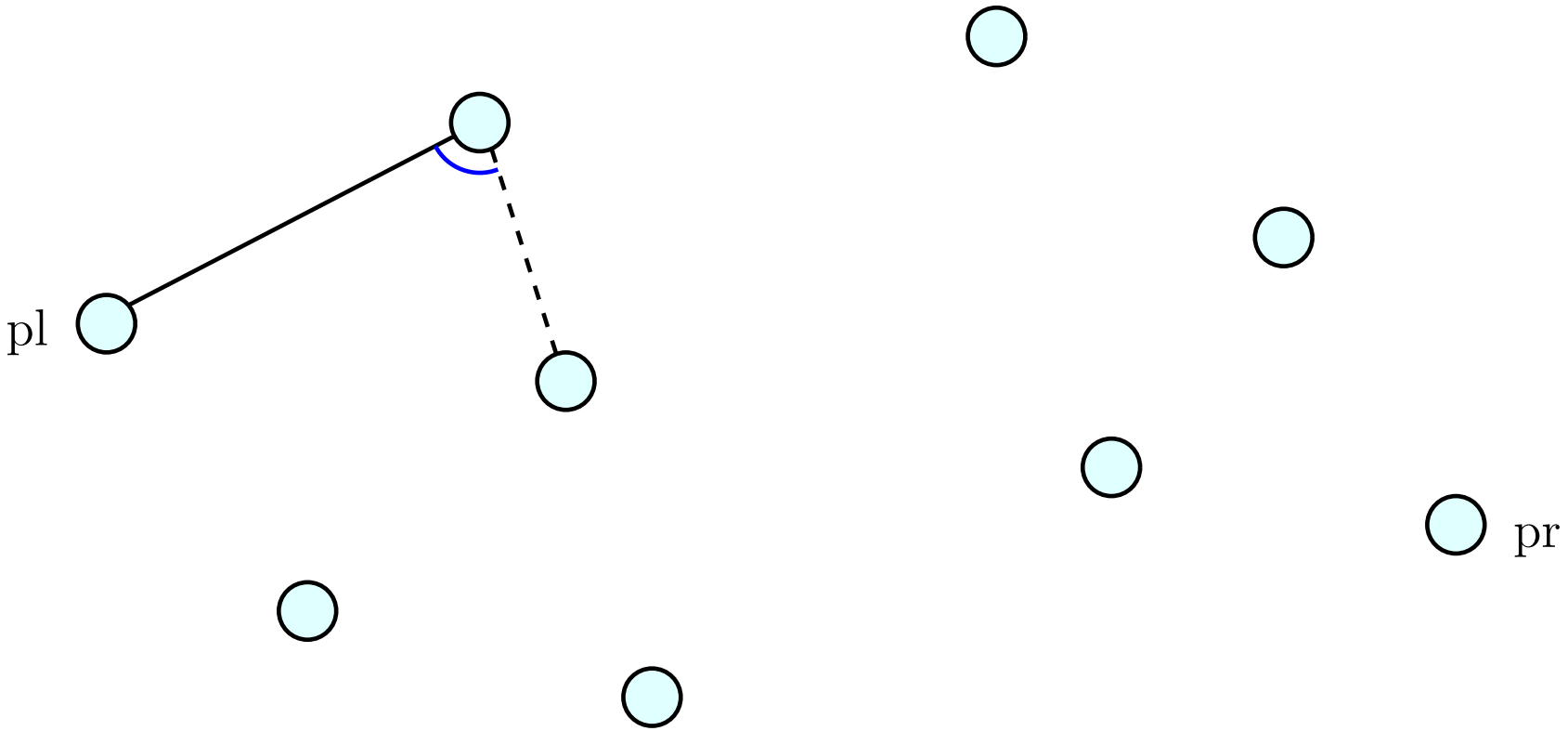
Grahamův algoritmus konstrukce konvexní obálky



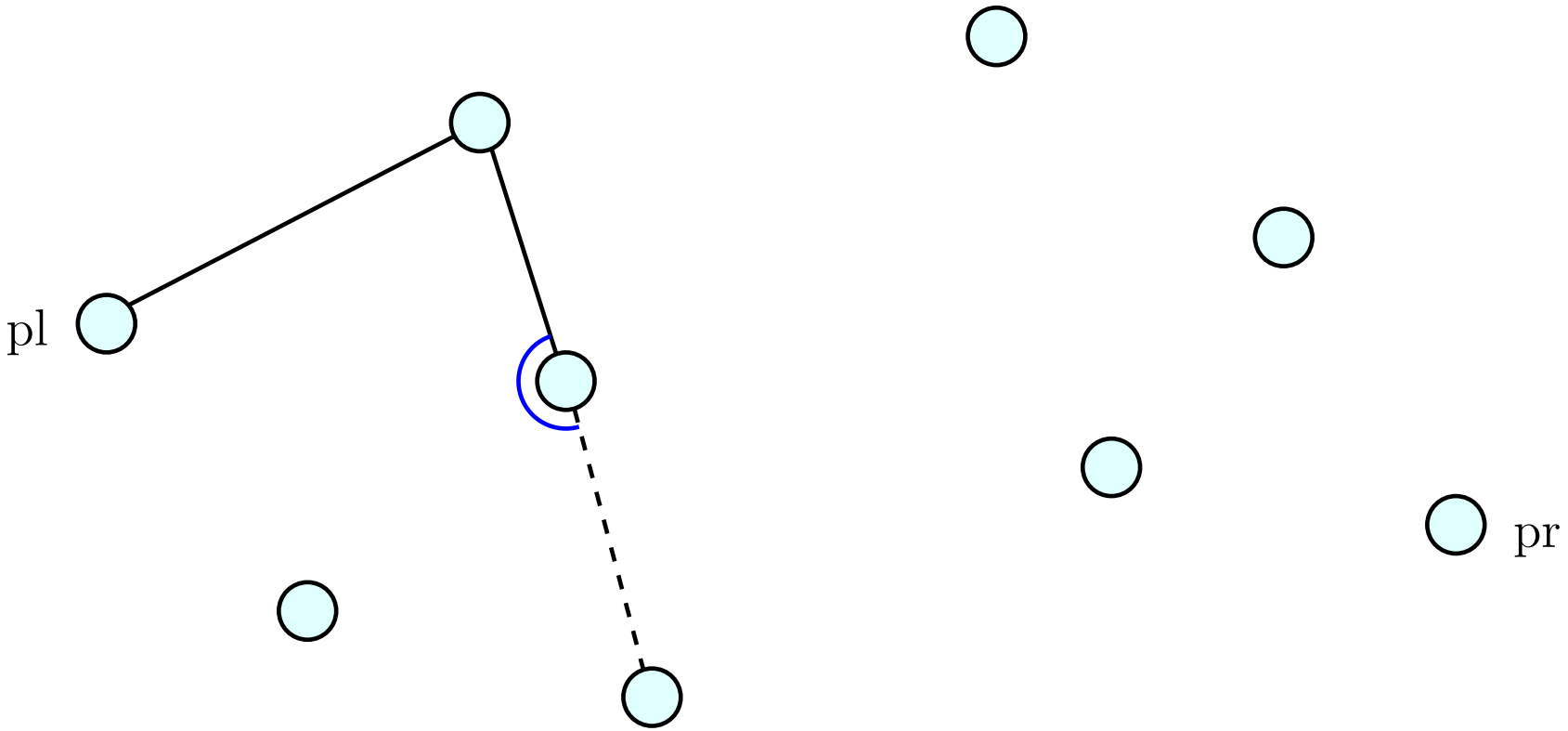
Grahamův algoritmus konstrukce konvexní obálky



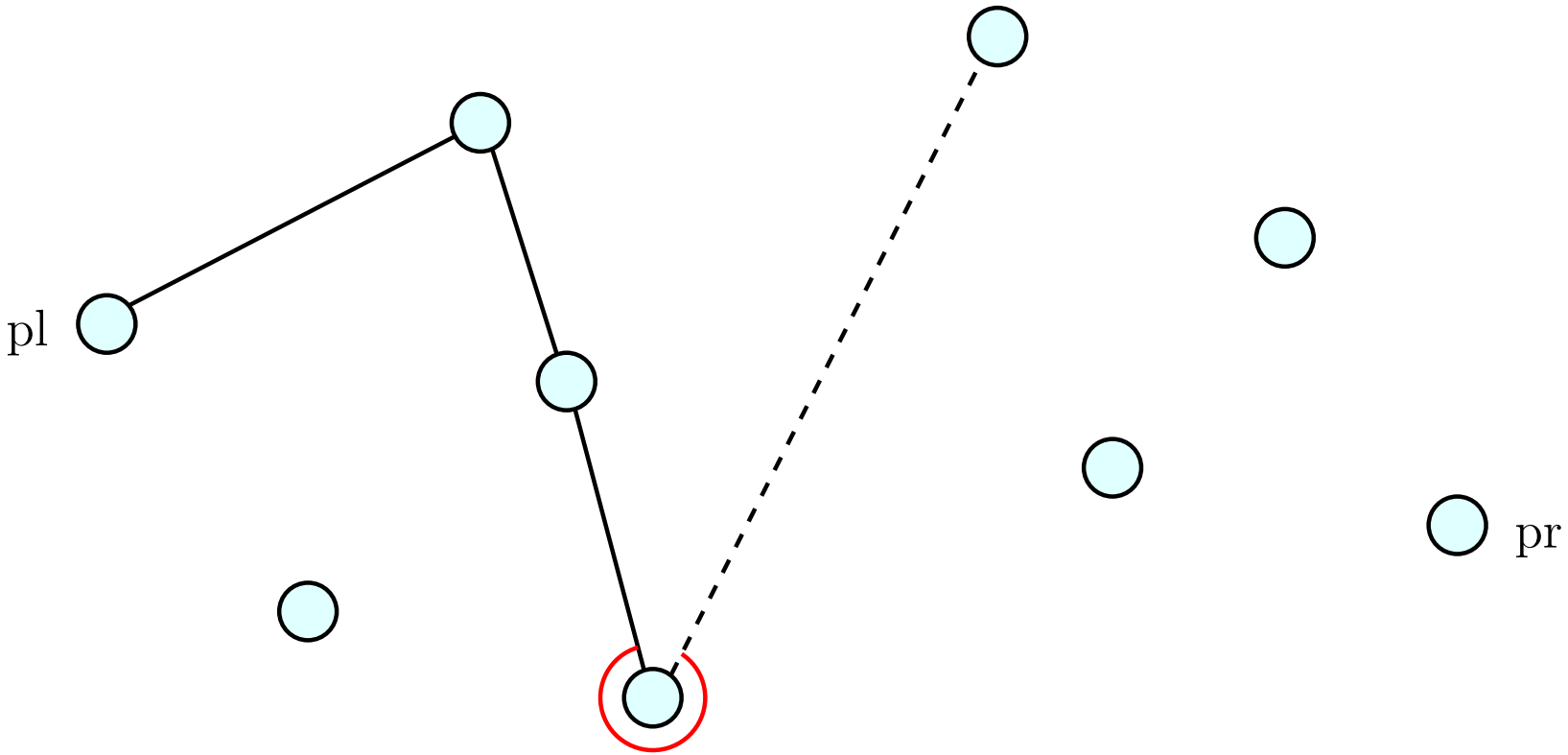
Grahamův algoritmus konstrukce konvexní obálky



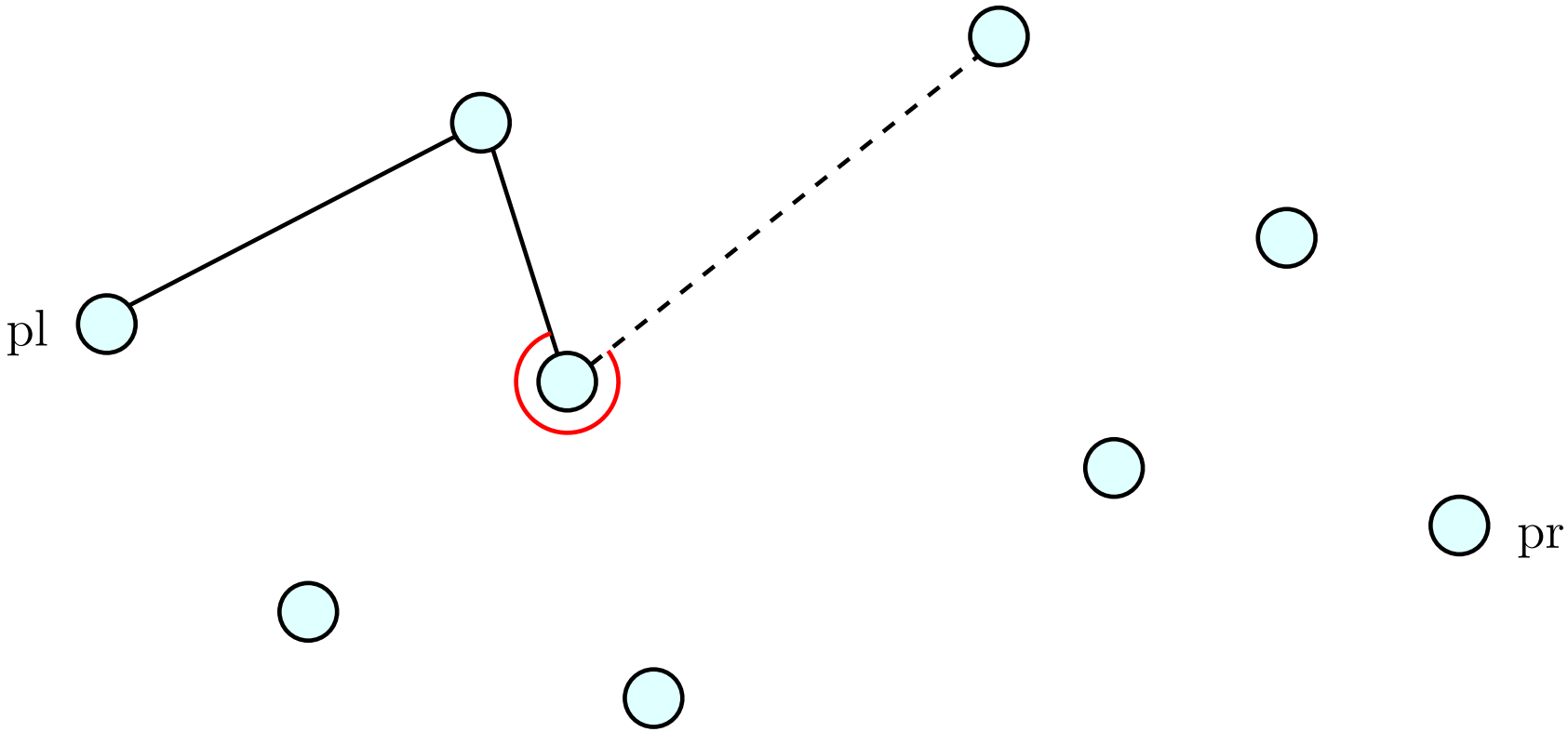
Grahamův algoritmus konstrukce konvexní obálky



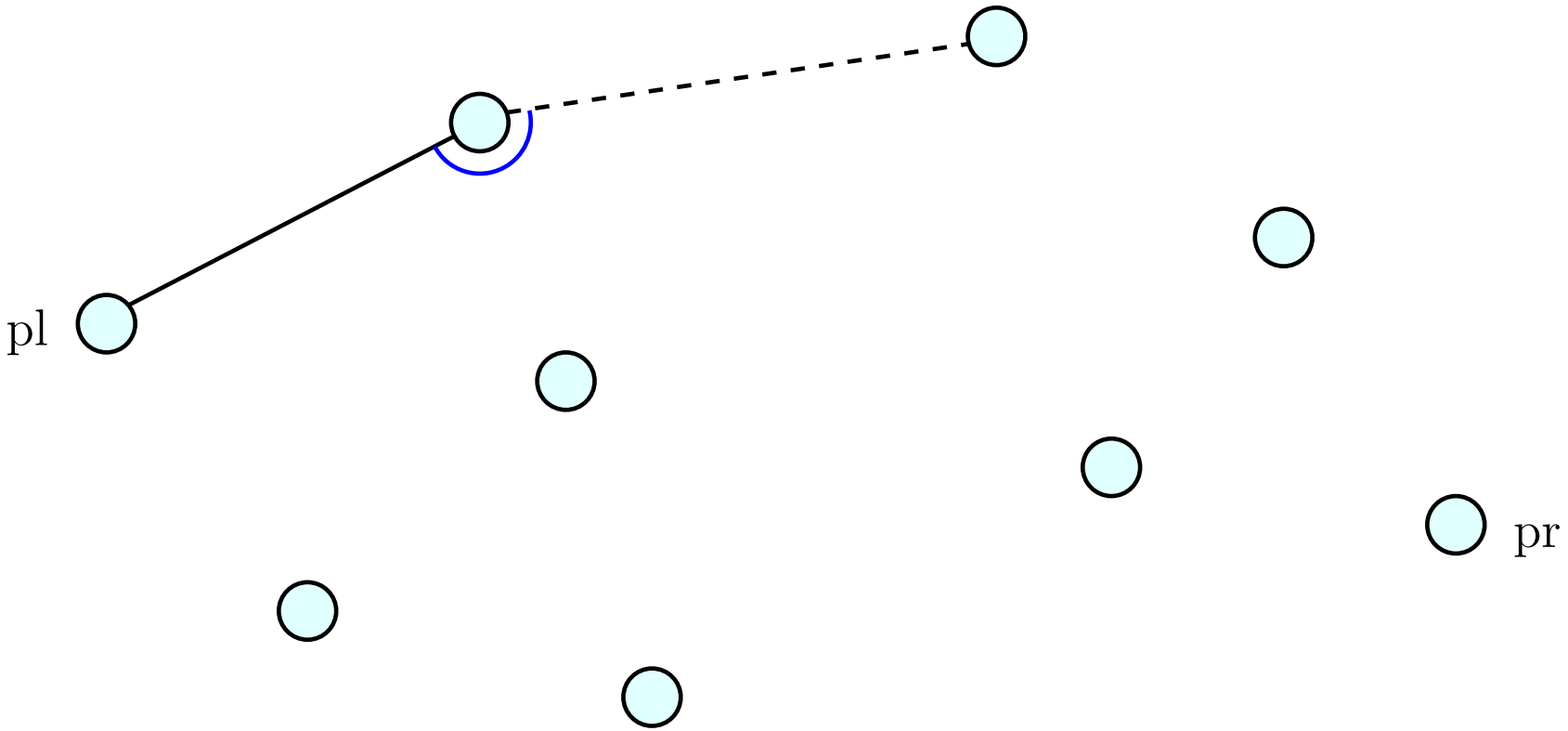
Grahamův algoritmus konstrukce konvexní obálky



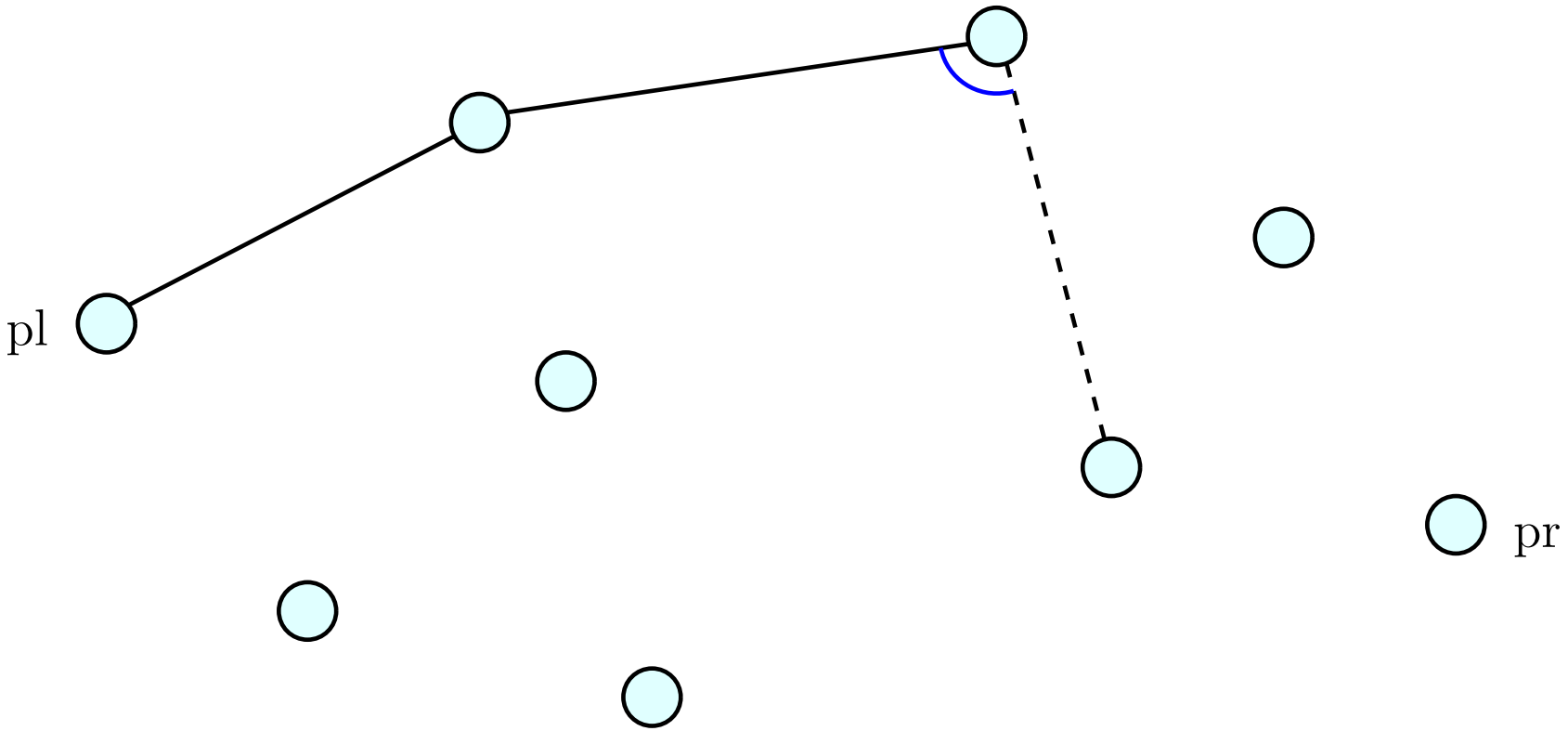
Grahamův algoritmus konstrukce konvexní obálky



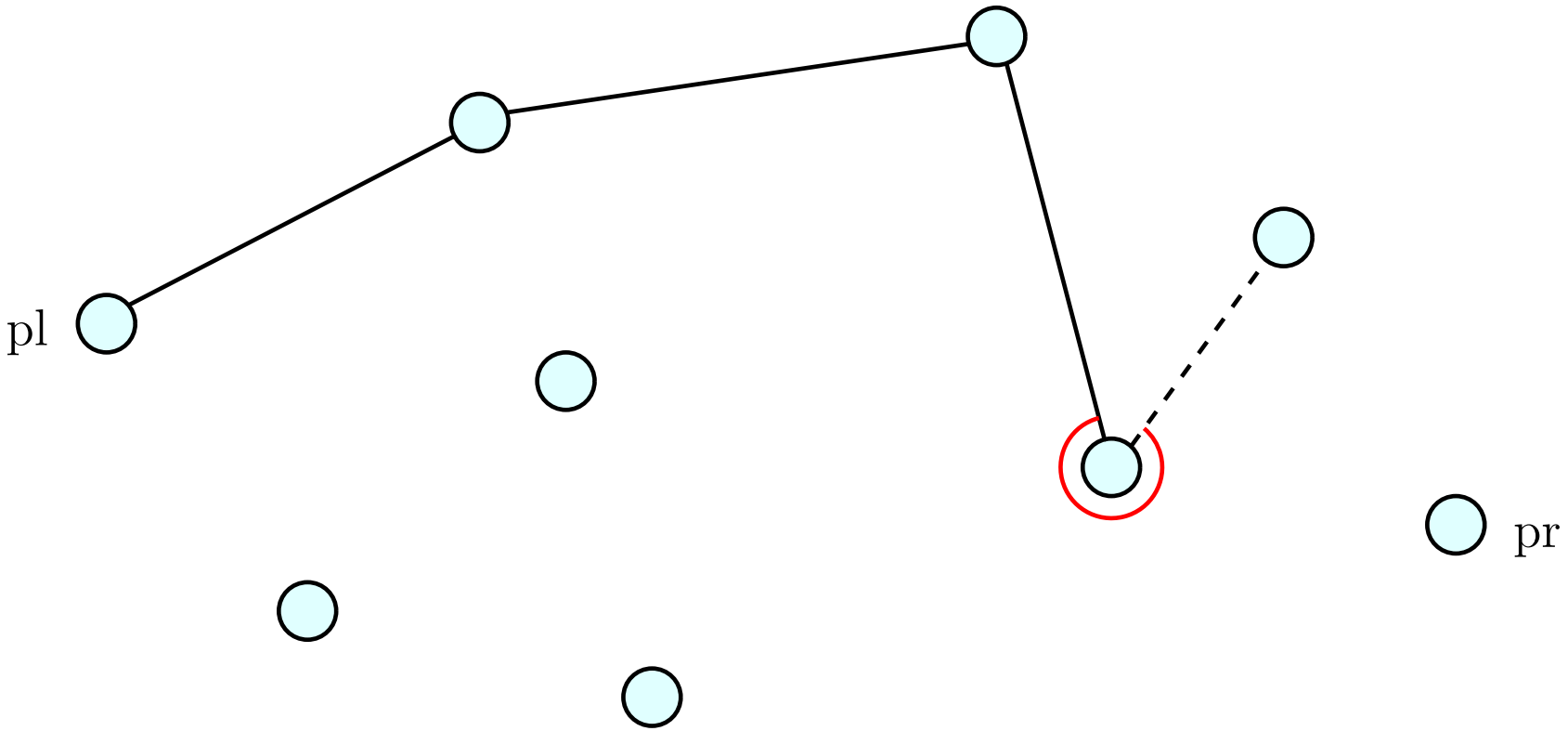
Grahamův algoritmus konstrukce konvexní obálky



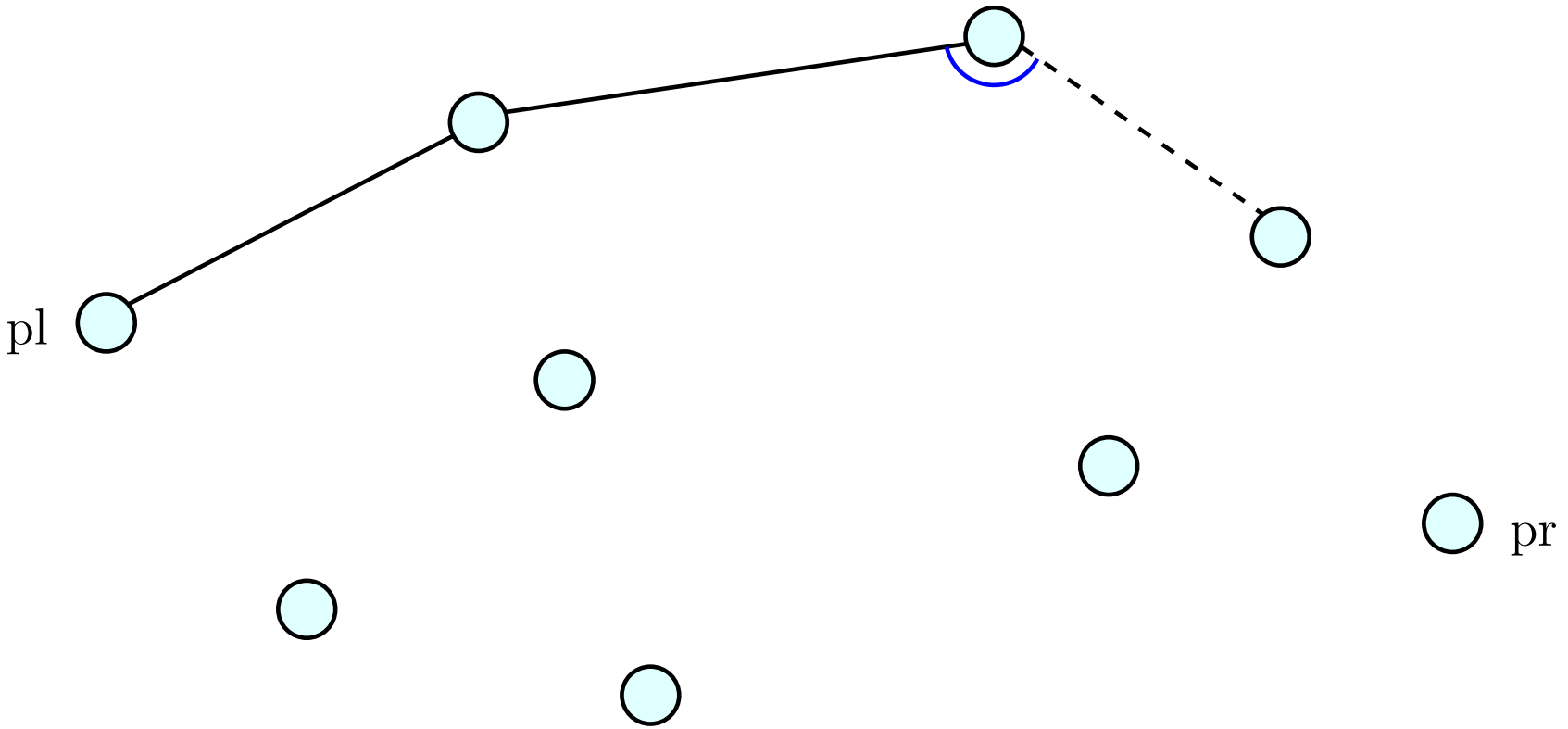
Grahamův algoritmus konstrukce konvexní obálky



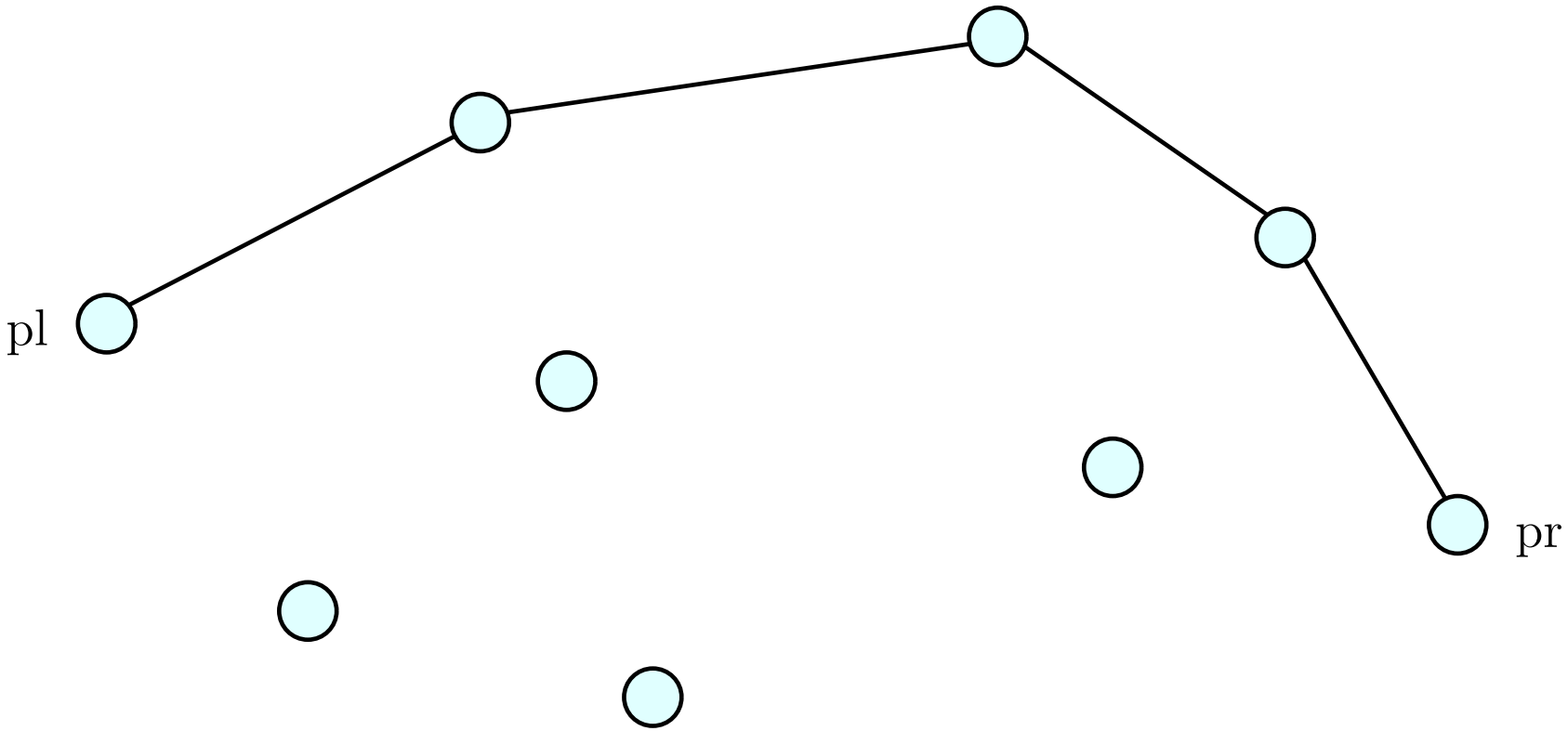
Grahamův algoritmus konstrukce konvexní obálky



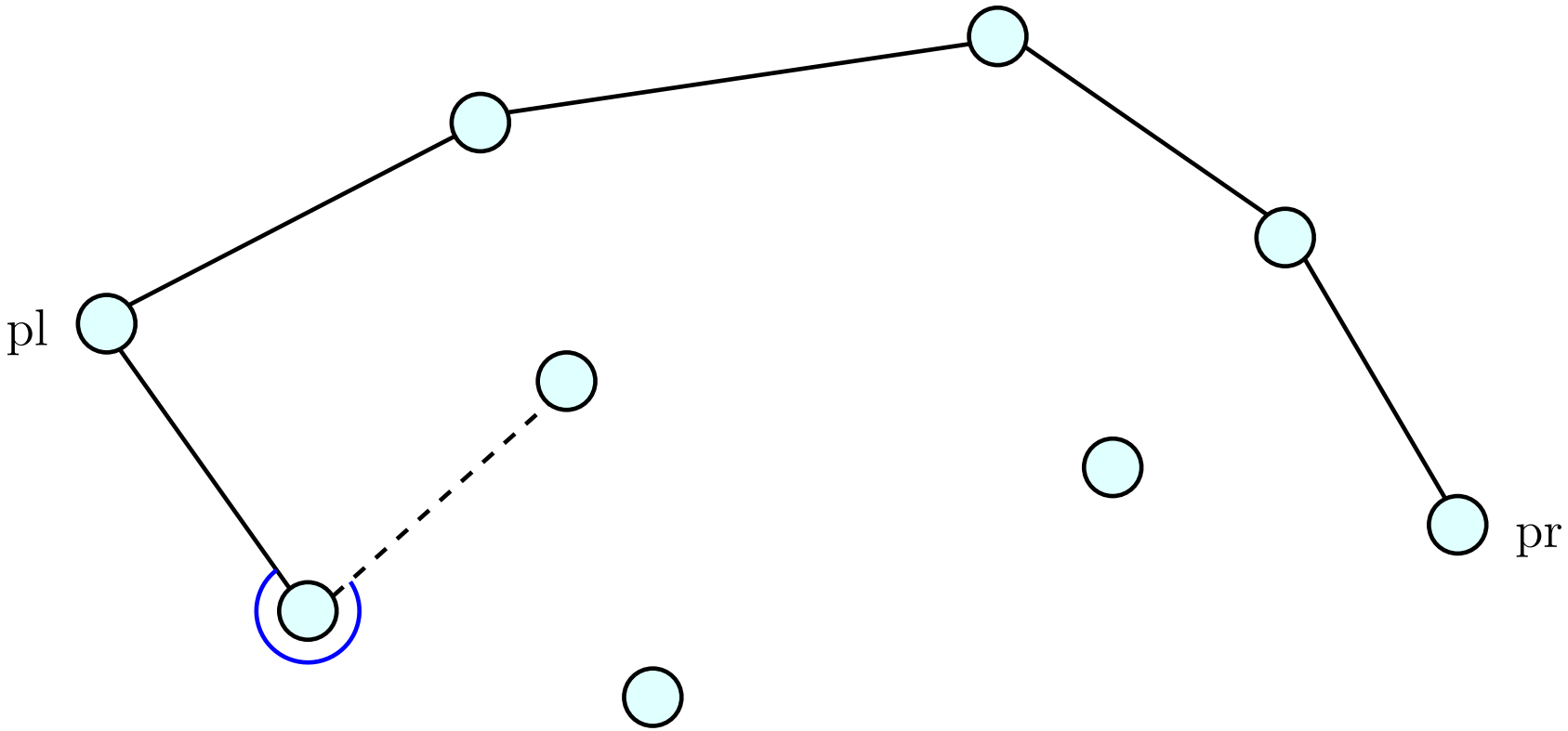
Grahamův algoritmus konstrukce konvexní obálky



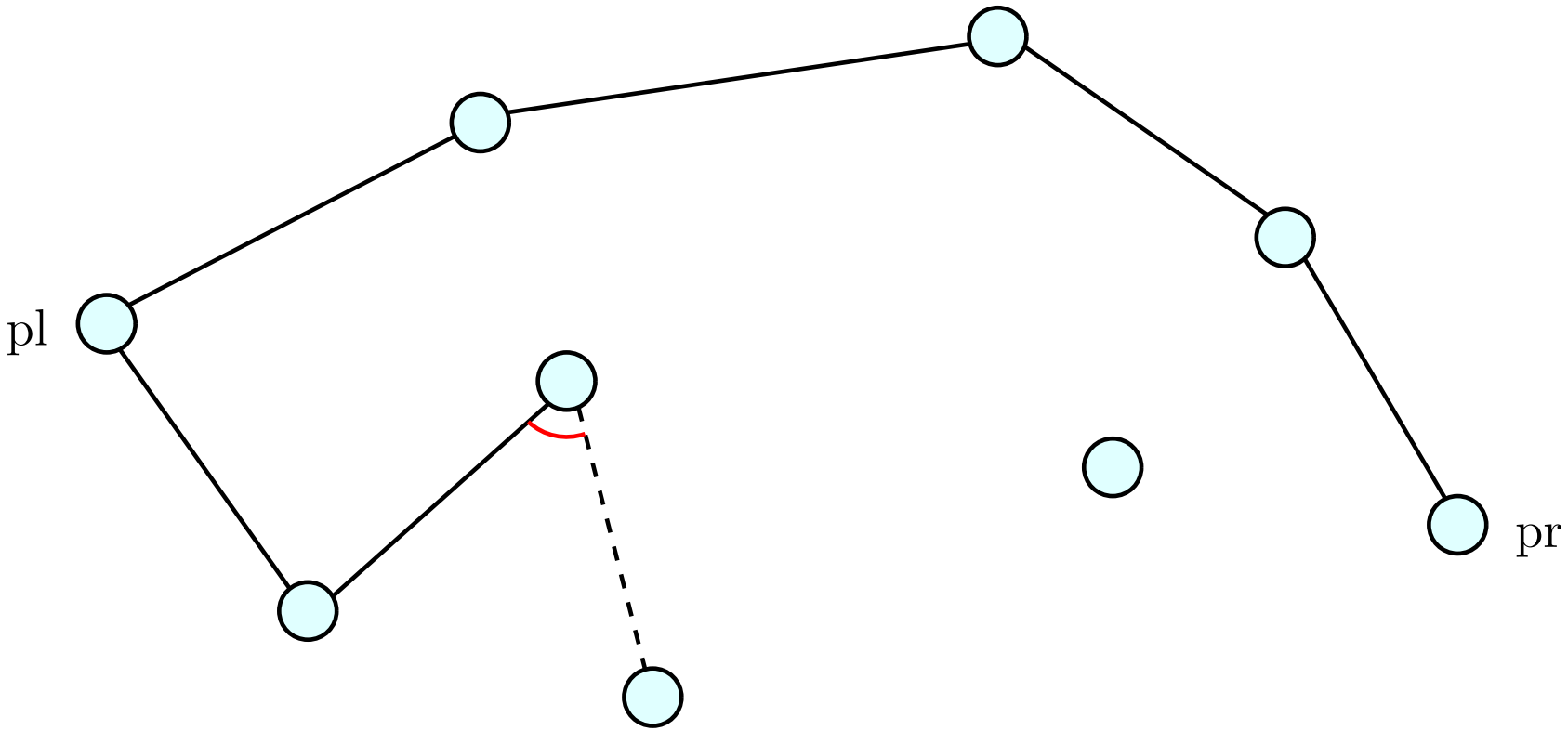
Grahamův algoritmus konstrukce konvexní obálky



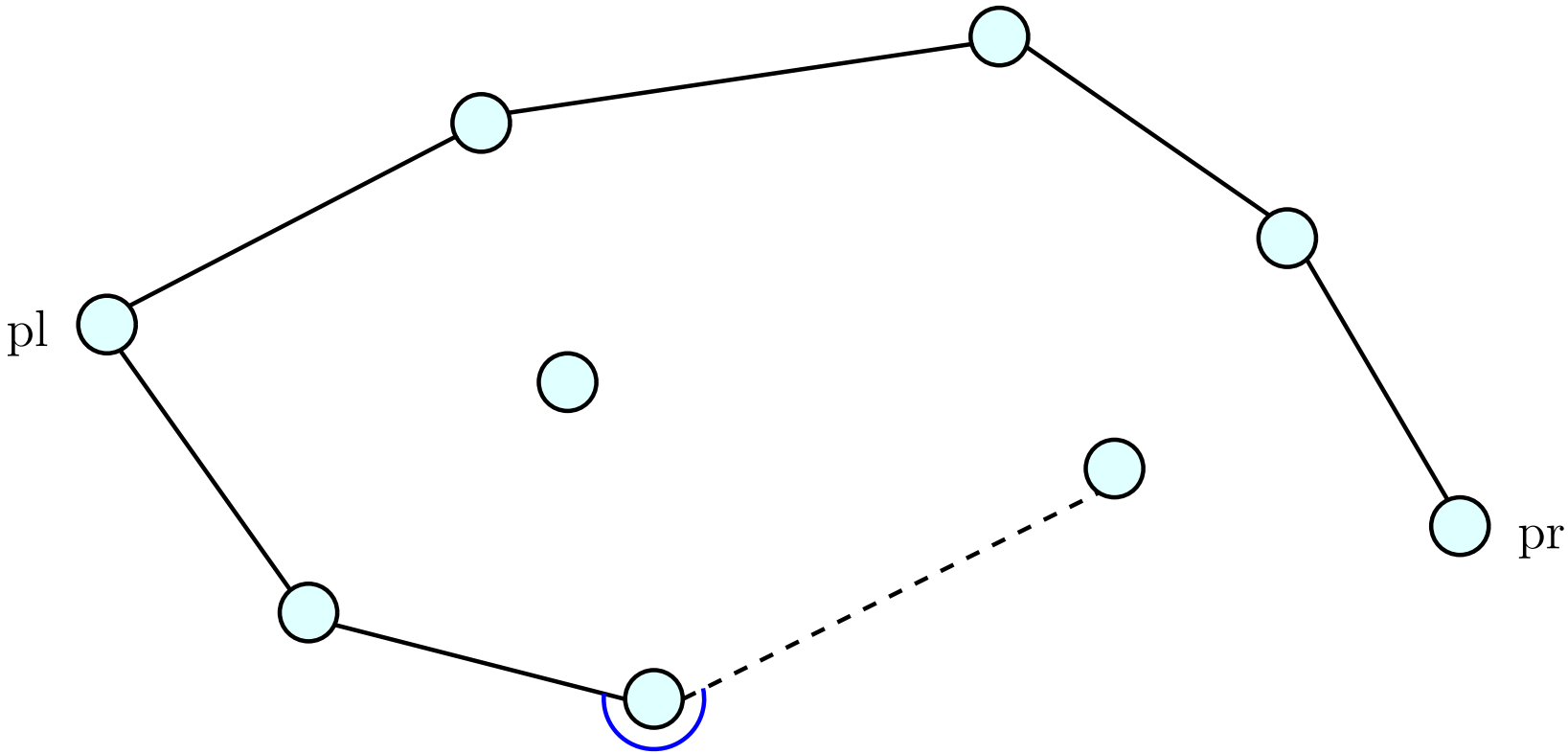
Grahamův algoritmus konstrukce konvexní obálky



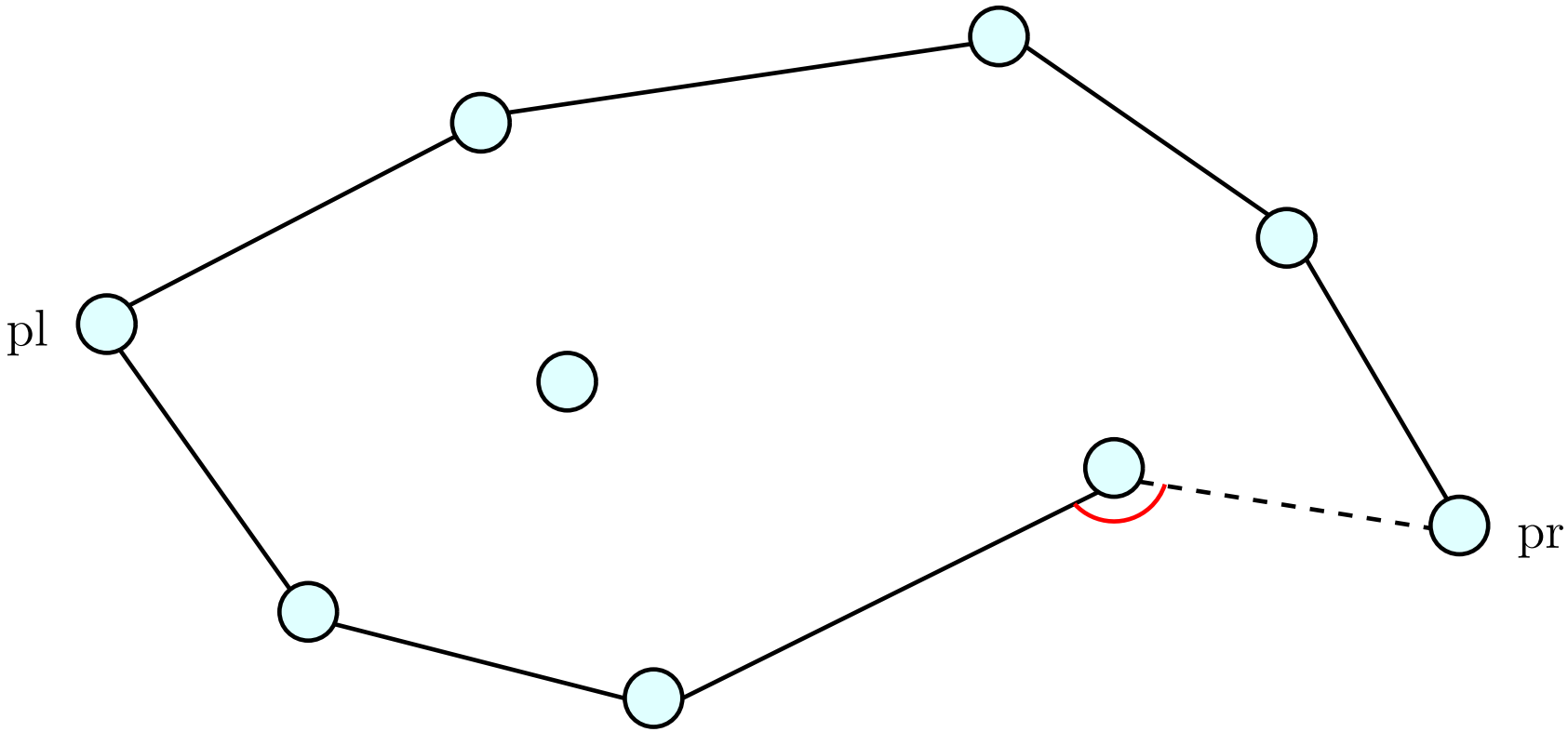
Grahamův algoritmus konstrukce konvexní obálky



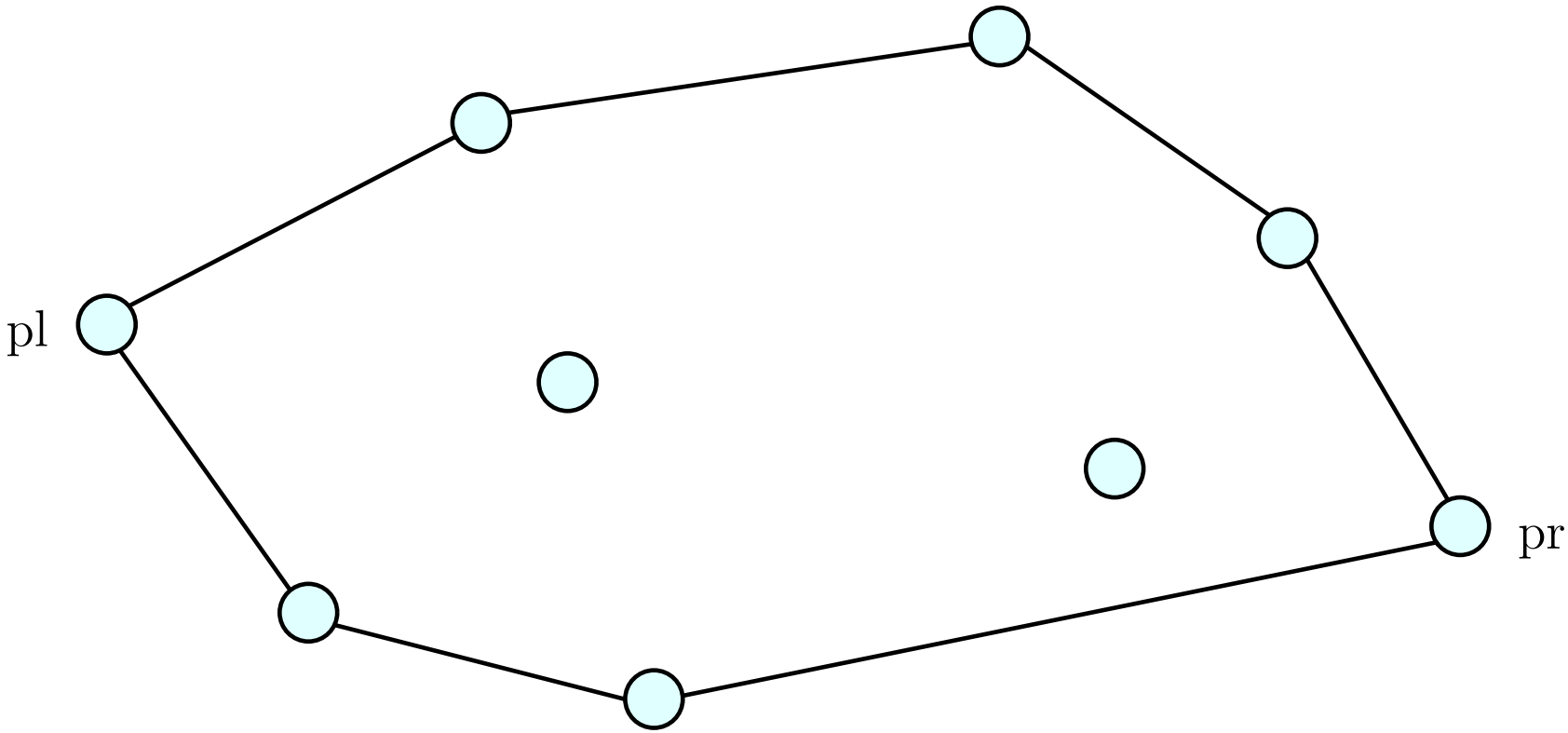
Grahamův algoritmus konstrukce konvexní obálky



Grahamův algoritmus konstrukce konvexní obálky



Grahamův algoritmus konstrukce konvexní obálky



Hledání konvexní obálky algorimem Rozděl a Panuj I

1. Body množiny se rozdělí na dvě stejně velké podmnožiny A a B tak, že všechny body množiny B mají větší x-ovou souřadnici,než bod z největší x-ovou souřadnicí z množiny A.
2. Rekurzivně je nalezena konvexní obálka obou množin A a B.
3. Konvexní obálky obálku nad oběma množinami A a B spojíme nalezením horní a dolní tečny a odstraněním všech uzlů, které jsou mezi těmito tečnami.

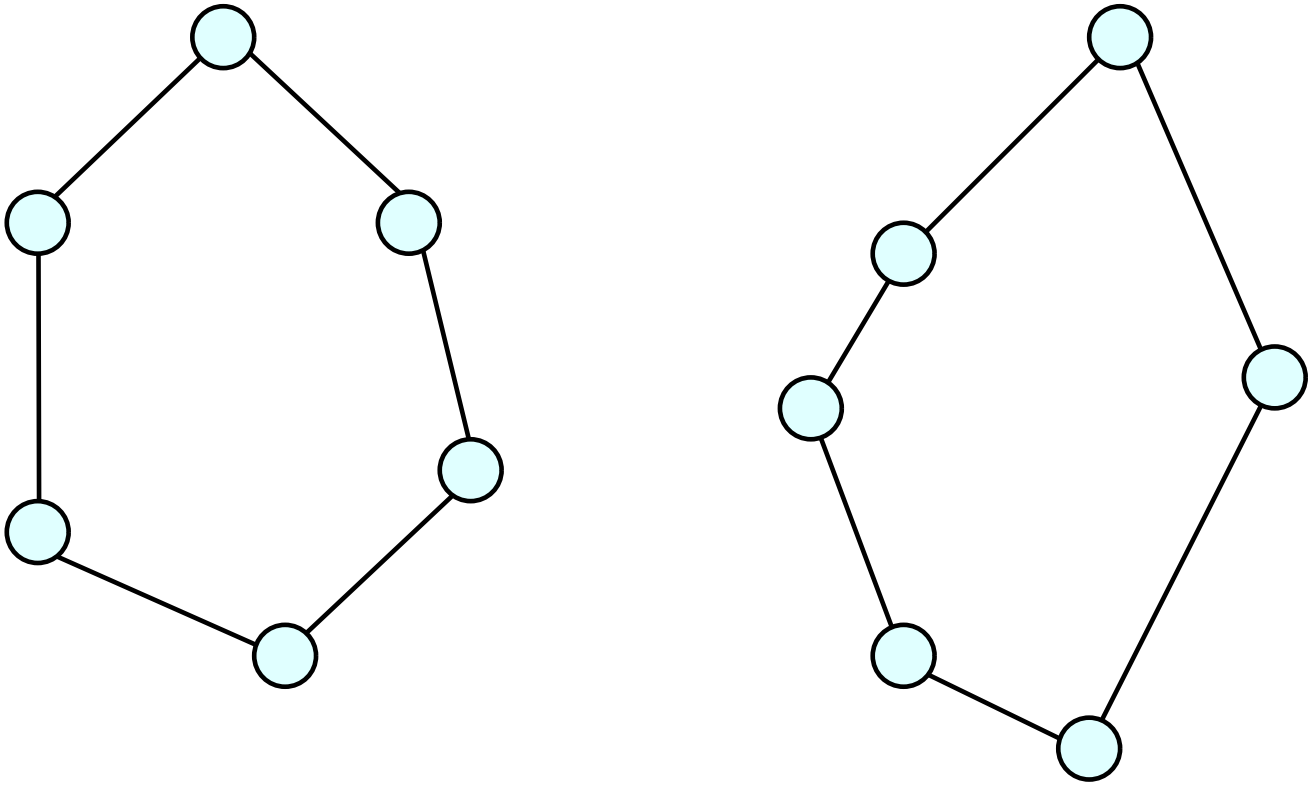
Hledání konvexní obálky algorimem Rozděl a Panuj II

Nalezení horní tečny mezi konvexními obálkami:

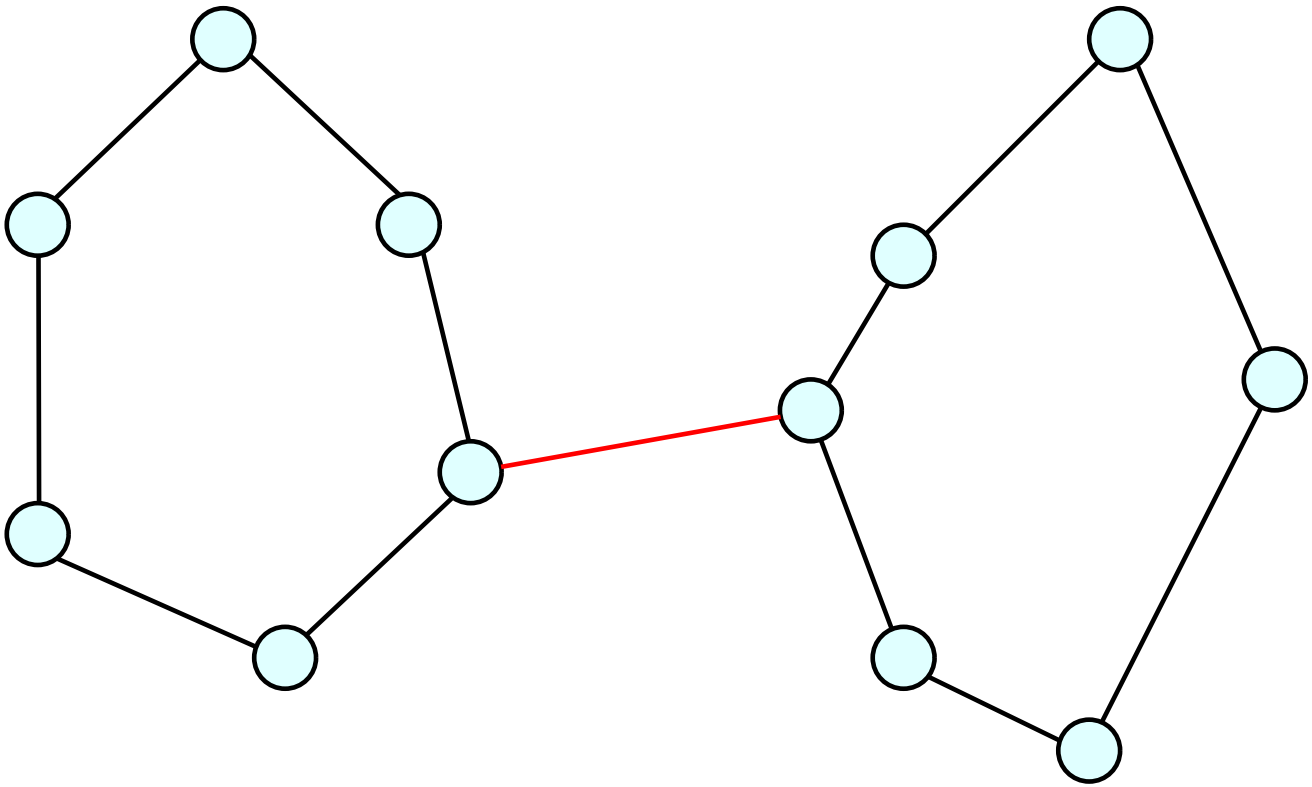
1. Bod a je bod s největší x-ovou souřadnicí z konvexní obálky nad množinou A.
2. Bod b je bod s nejmenší x-ovou souřadnicí z konvexní obálky nad množinou B..
3. Dokud není ab horní tečnou prováděj
   - procházej seznam bodů reprezentující konvexní obálku z bodu a proti směru hodinových ručiček, dokud není ab horní tečnou konvexní obálky množiny A.
   - procházej seznam bodů reprezentující konvexní obálku z bodu b po směru hodinových ručiček, dokud není ab horní tečnou konvexní obálky množiny B.

Nalezení dolní tečny je obdobné.

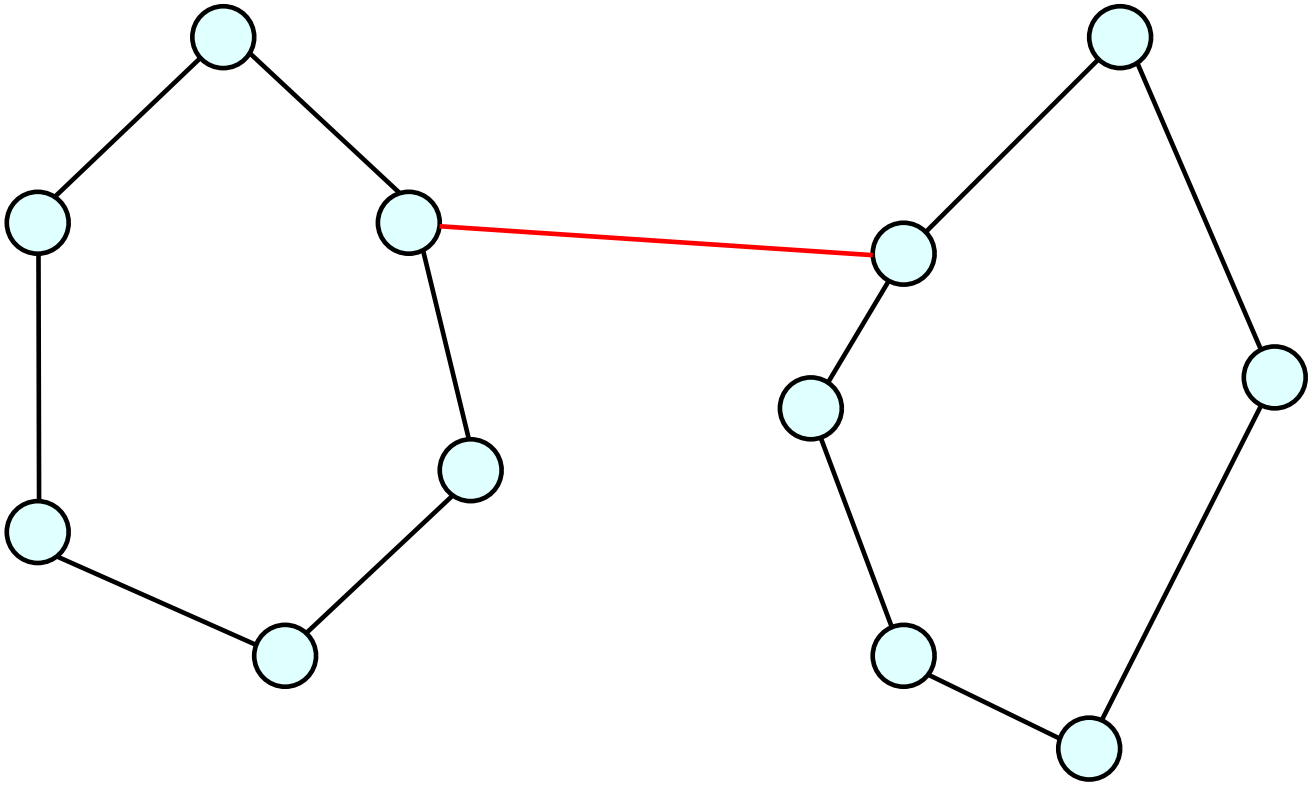
Hledání konvexní obálky Rozdel a Panuj priklad



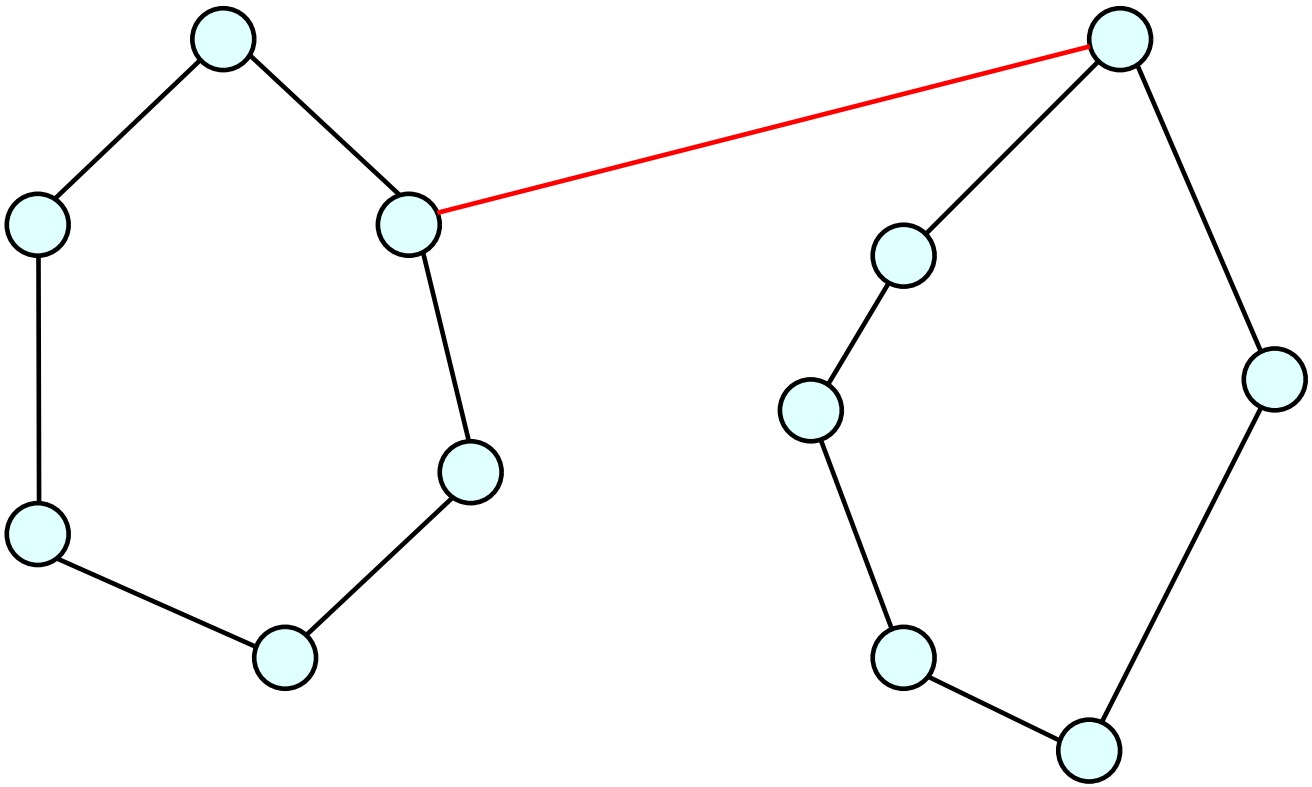
Hledání konvexní obálky Rozdel a Panuj priklad



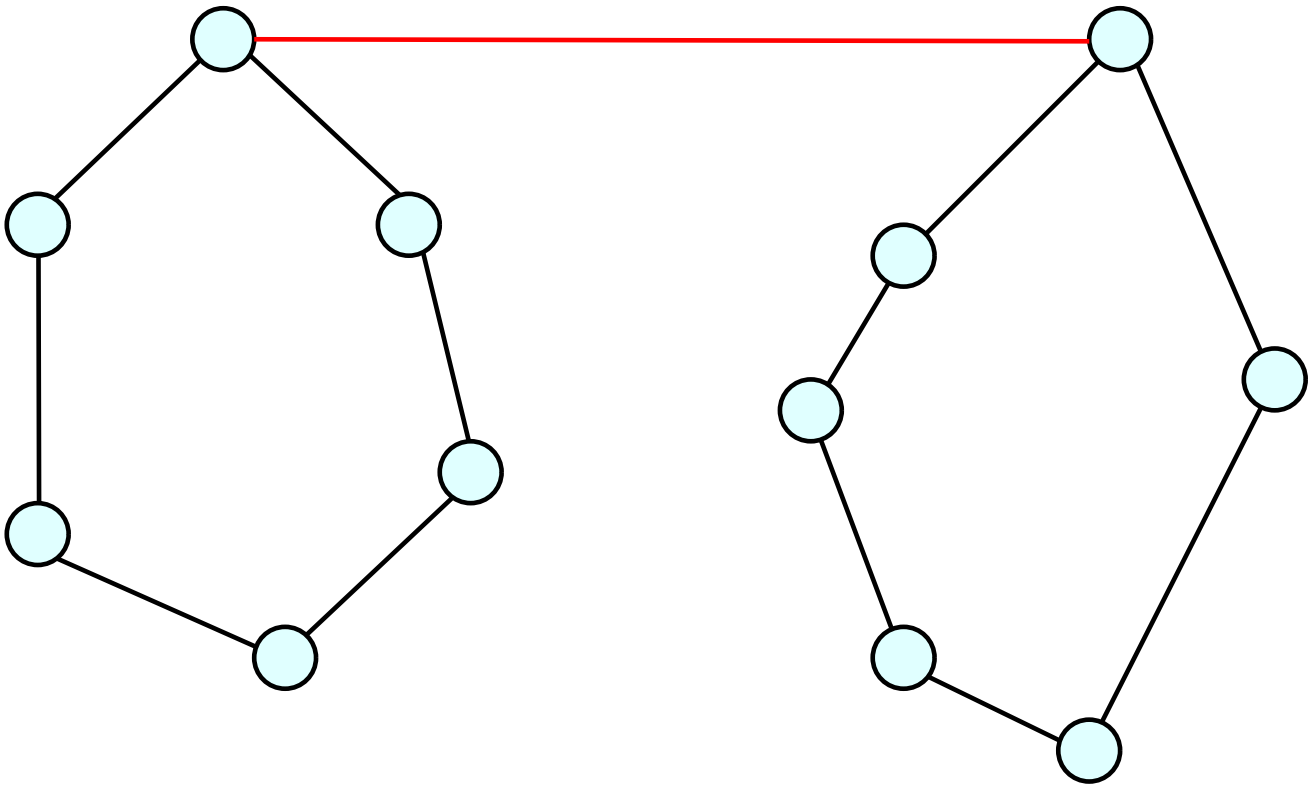
Hledání konvexní obálky Rozdel a Panuj priklad



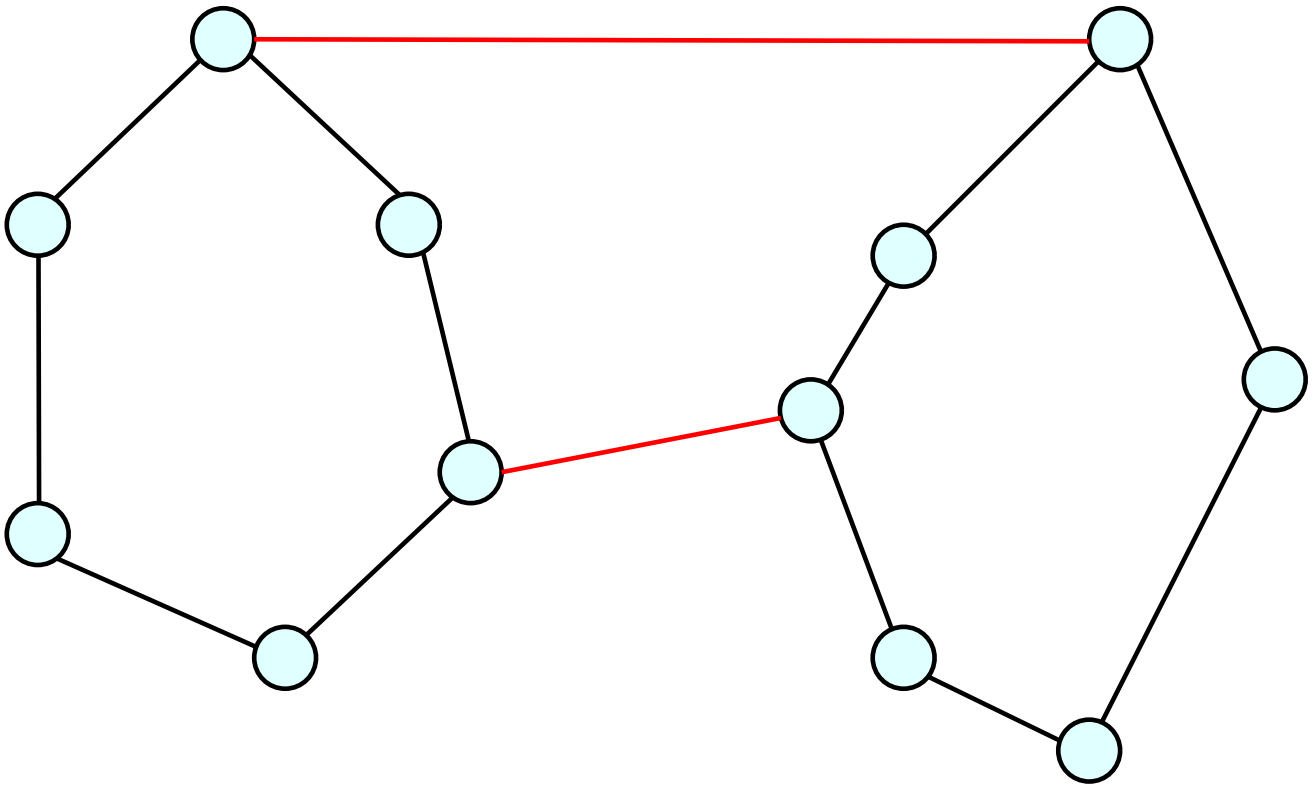
Hledání konvexní obálky Rozdel a Panuj priklad



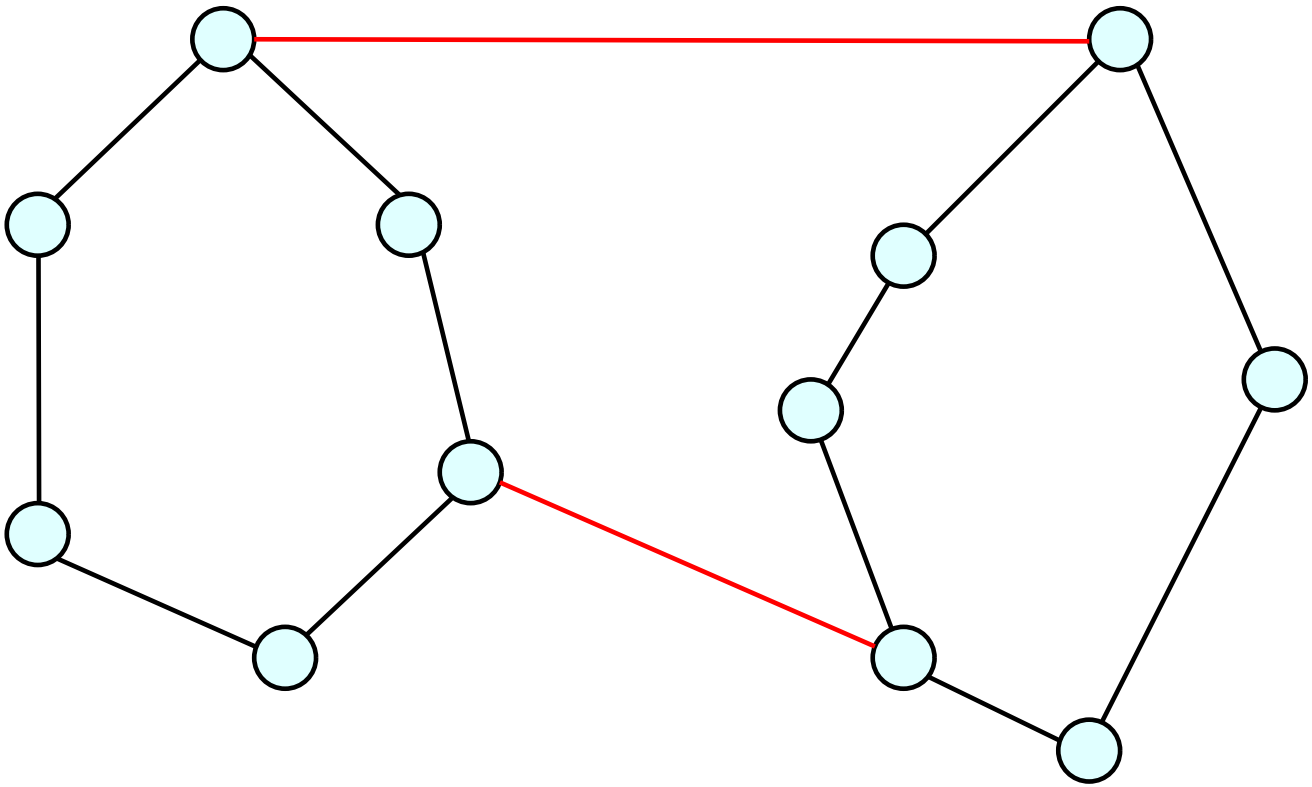
Hledání konvexní obálky Rozdel a Panuj priklad



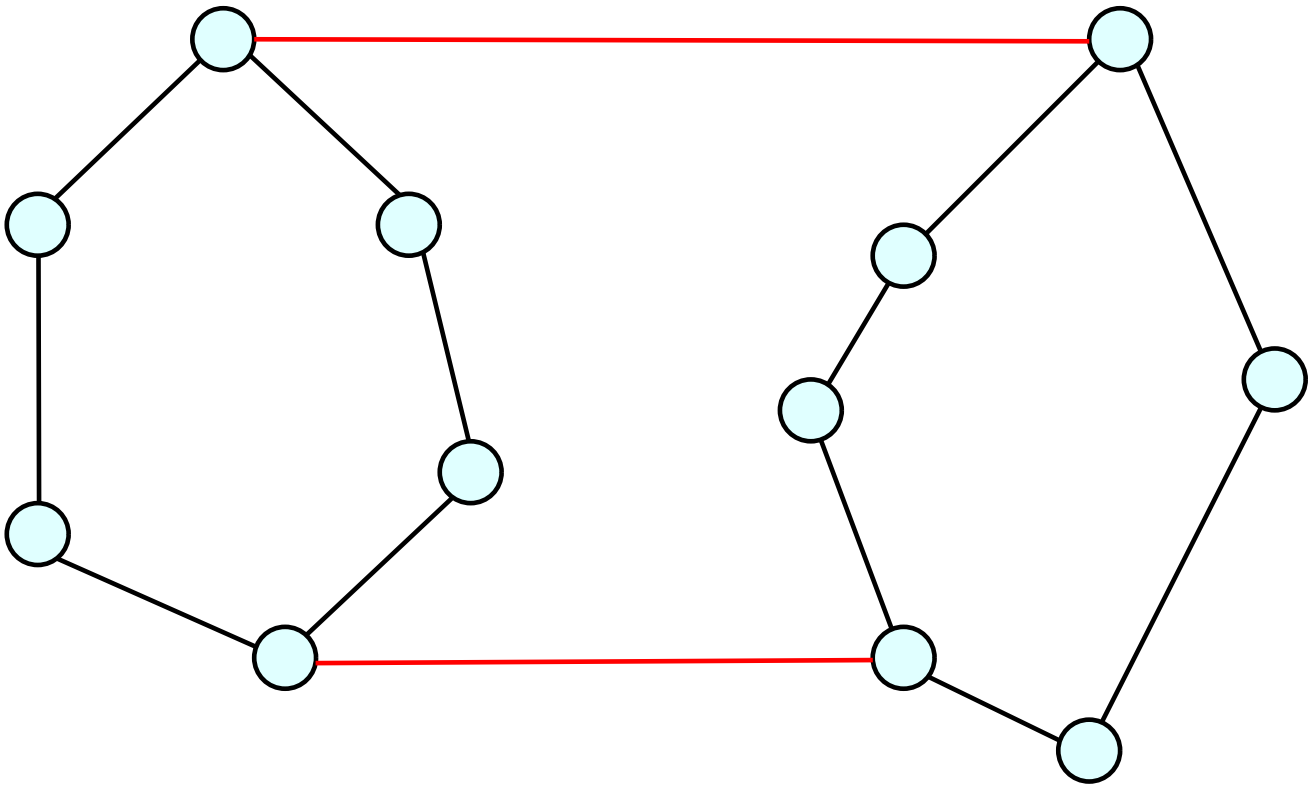
Hledání konvexní obálky Rozdel a Panuj priklad



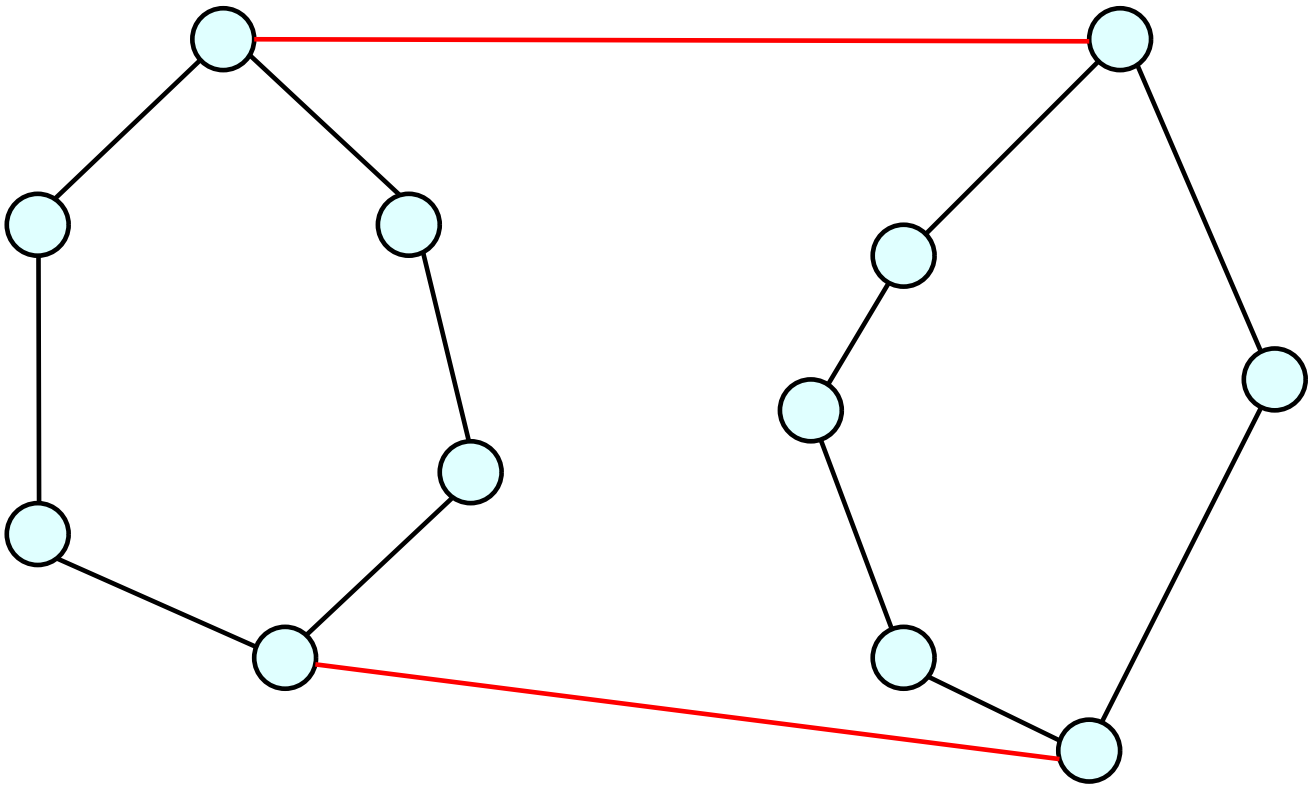
Hledání konvexní obálky Rozdel a Panuj priklad



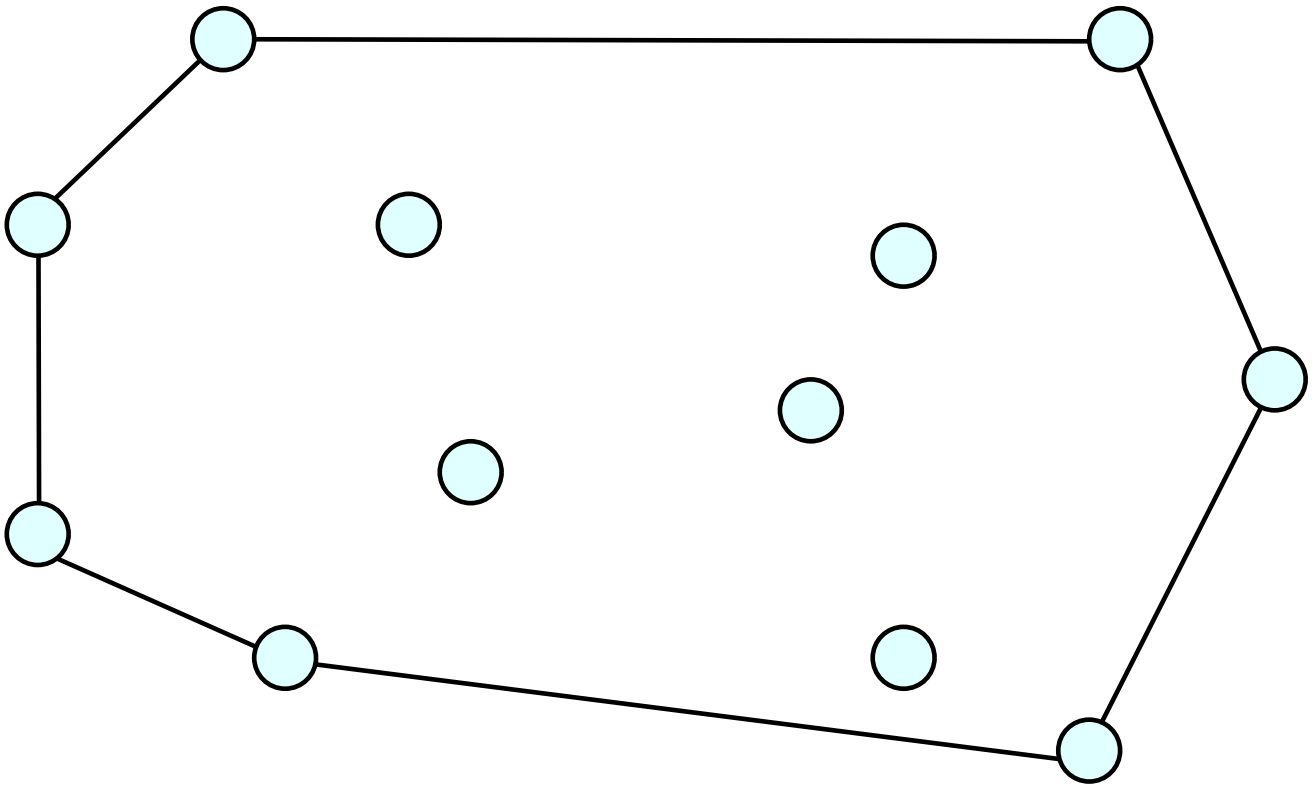
Hledání konvexní obálky Rozdel a Panuj priklad



Hledání konvexní obálky Rozdel a Panuj priklad



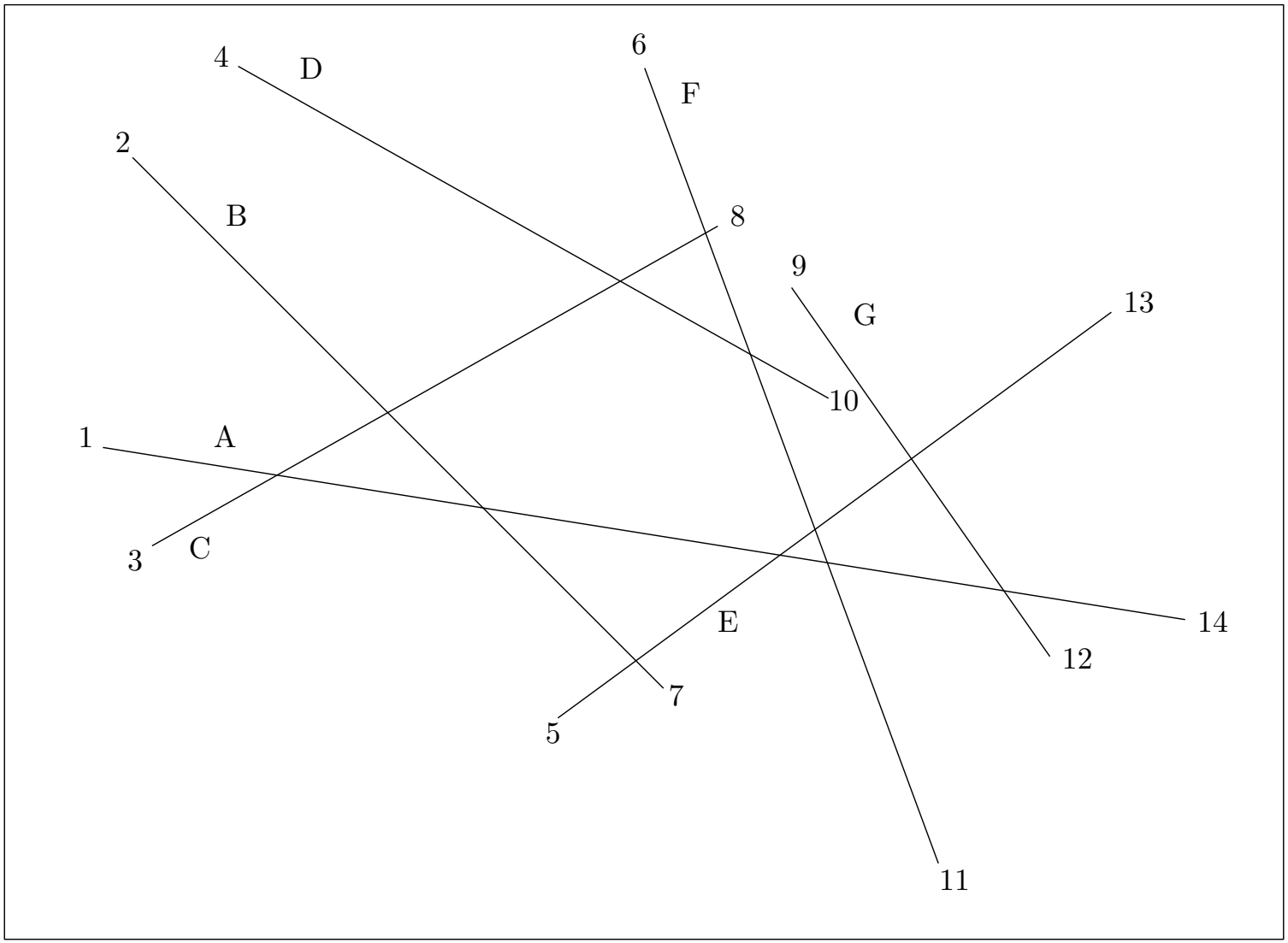
Hledání konvexní obálky Rozdel a Panuj priklad



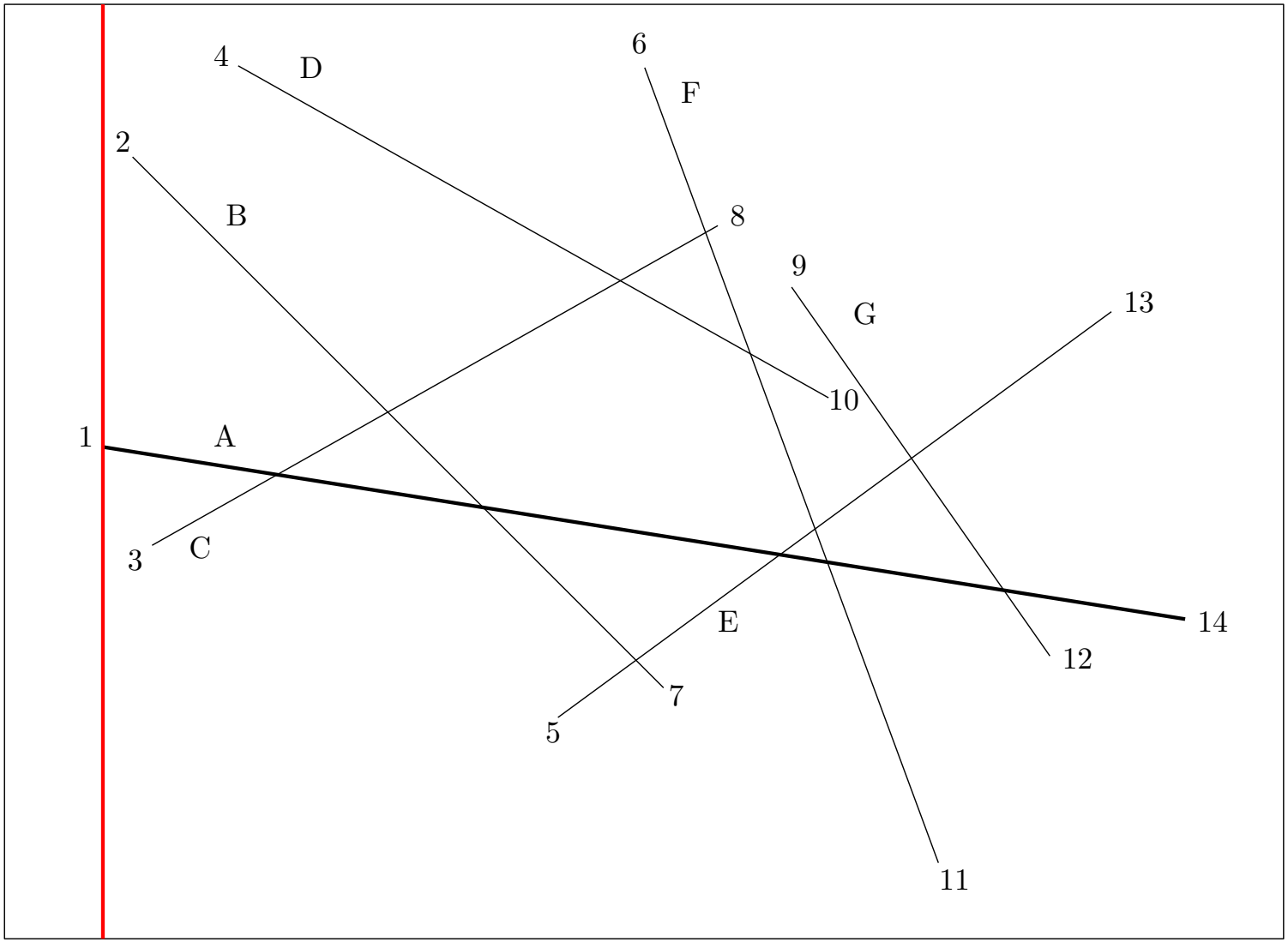
Zametací hyperrovina

1. Vytvoření X-struktury - počáteční a koncové body úseček seřazené podle x.
2. Vytvoření Y-struktury - seznam úseček v řezu zametací roviny seřazené podle y (na začítku je prázdná)
3. Dokud není x-struktura prázdná prováděj:
   - Jdi na první bod x-struktury, pokud je počáteční, přidej přímku do Y-struktury, pokud koncový odeber ji. Pokud průsečík přímky prohoď.
   - Prozkoumej nově vzniklé sousedy v y-struktuře a urči potenciální průsečíky na pravo od zametací přímky
   - odeber zpracovaný bod z x-struktury

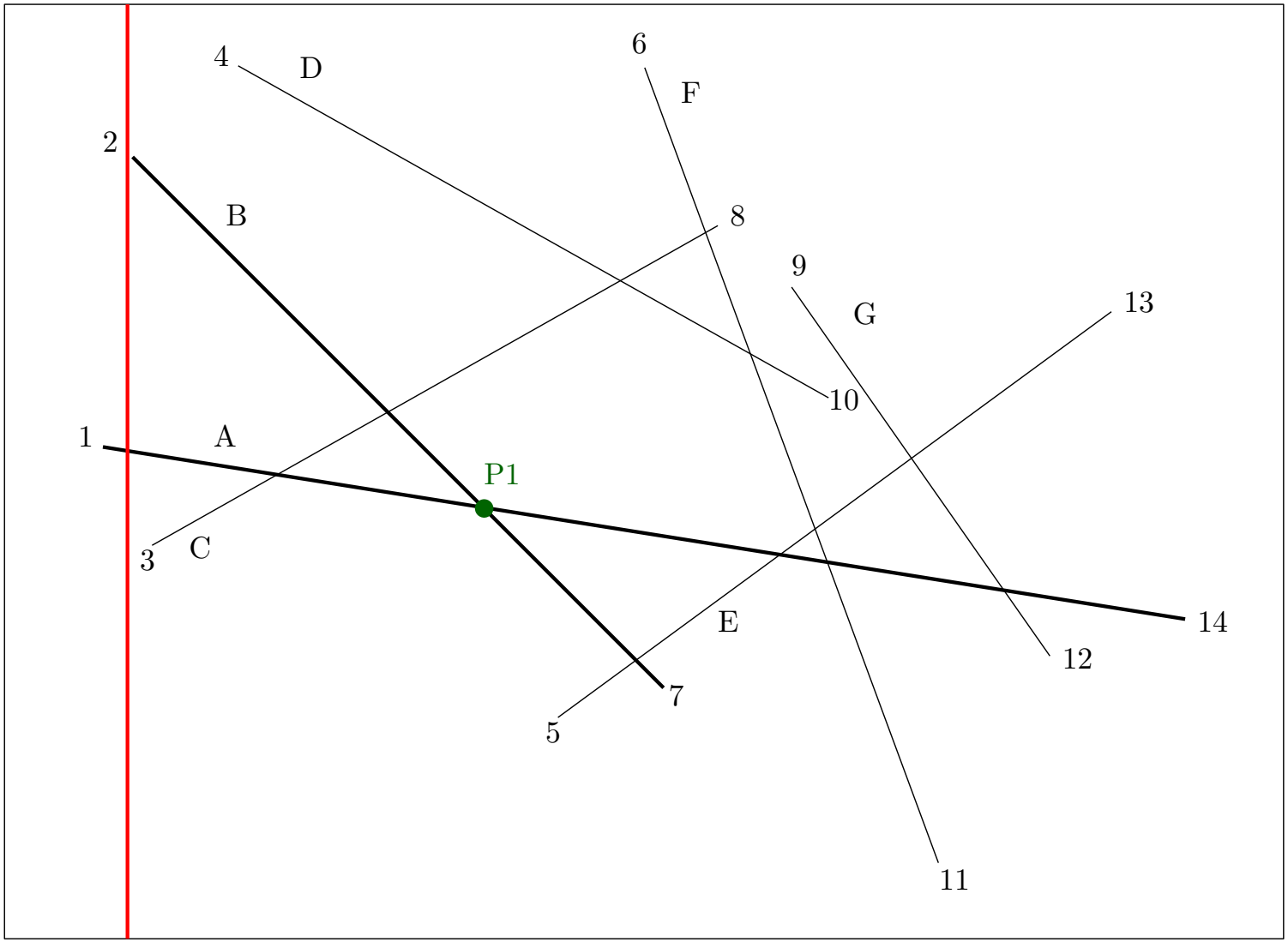
Zametací hyperrovina



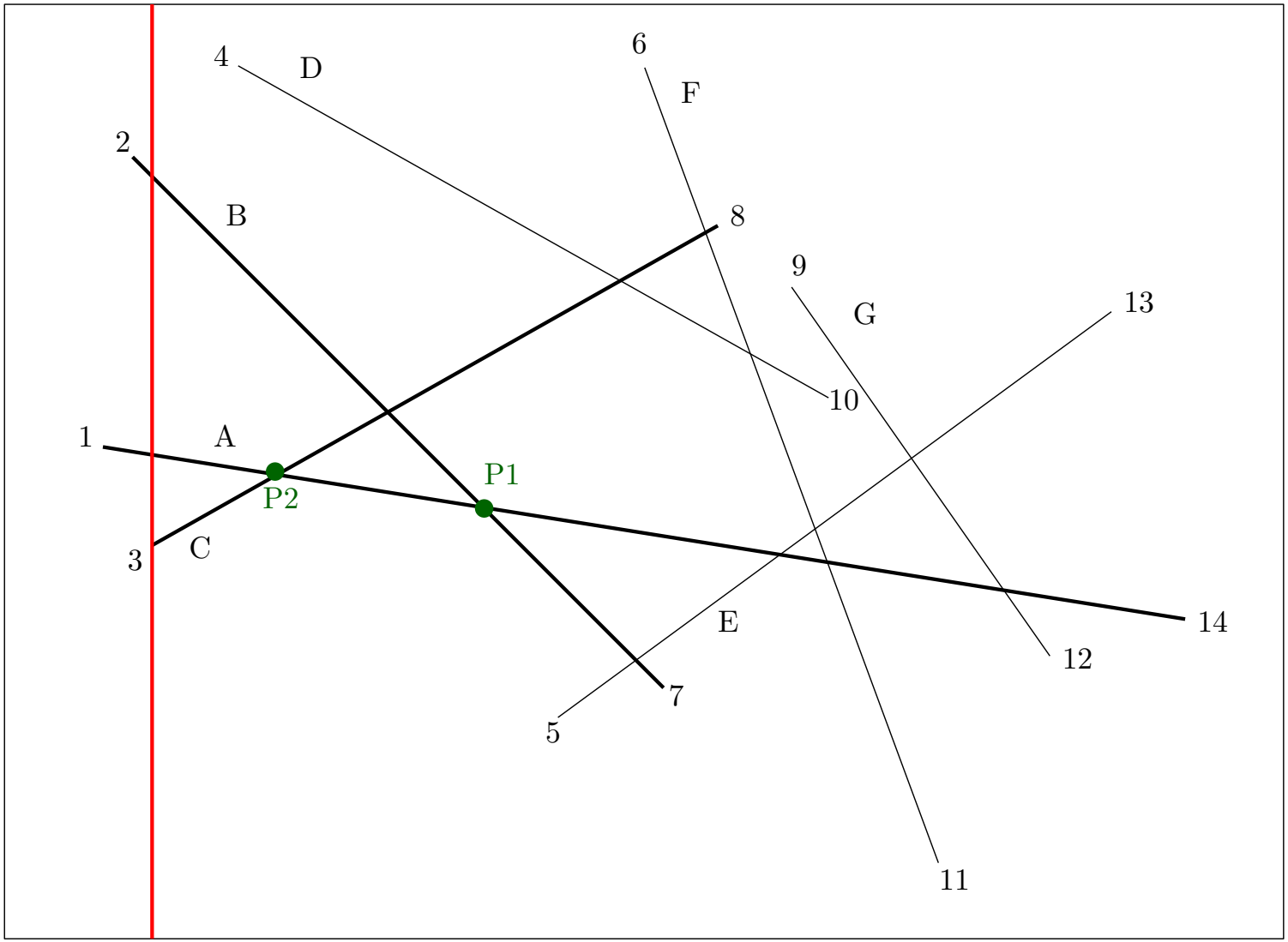
Zametací hyperrovina



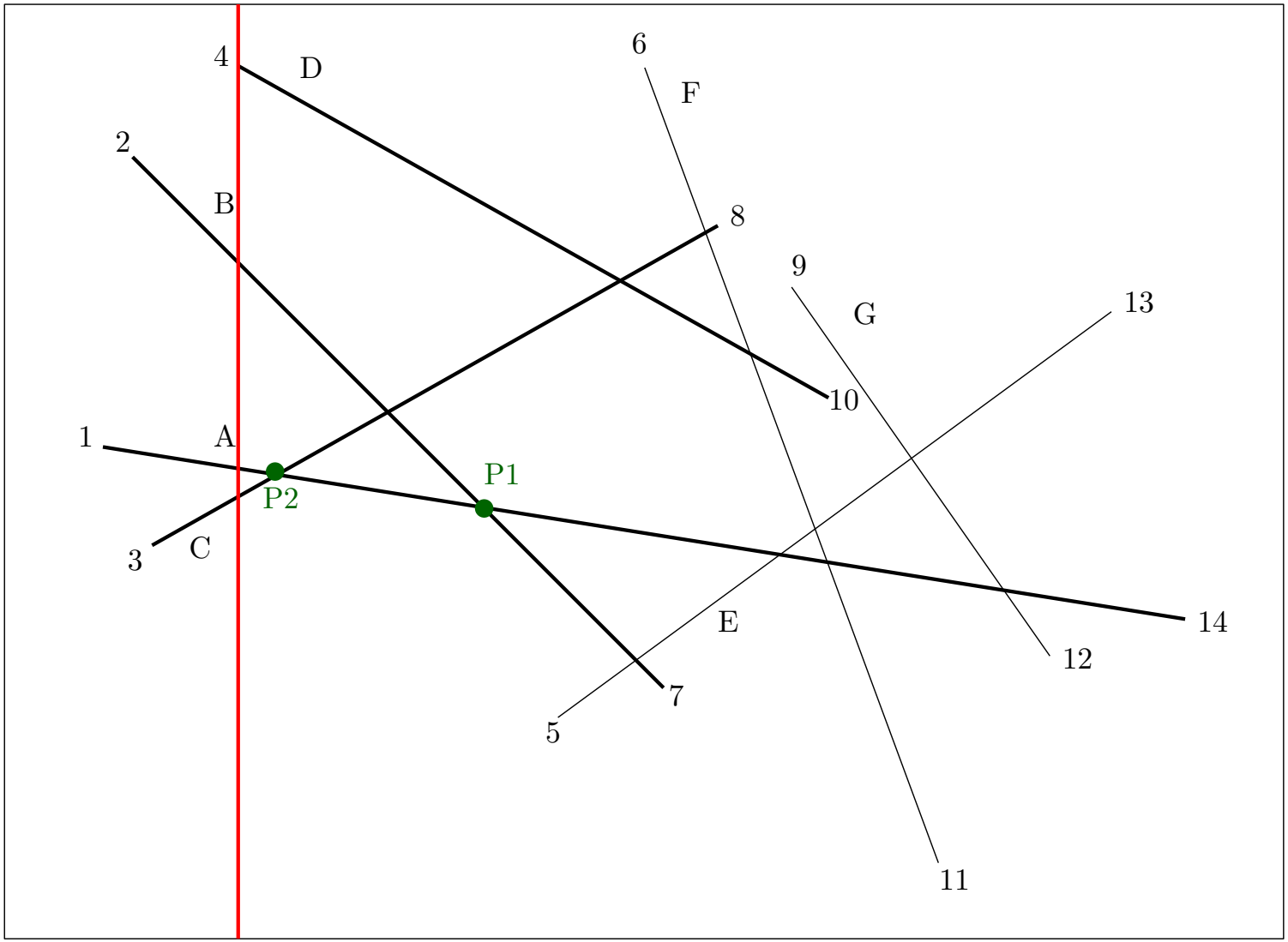
Zametací hyperrovina



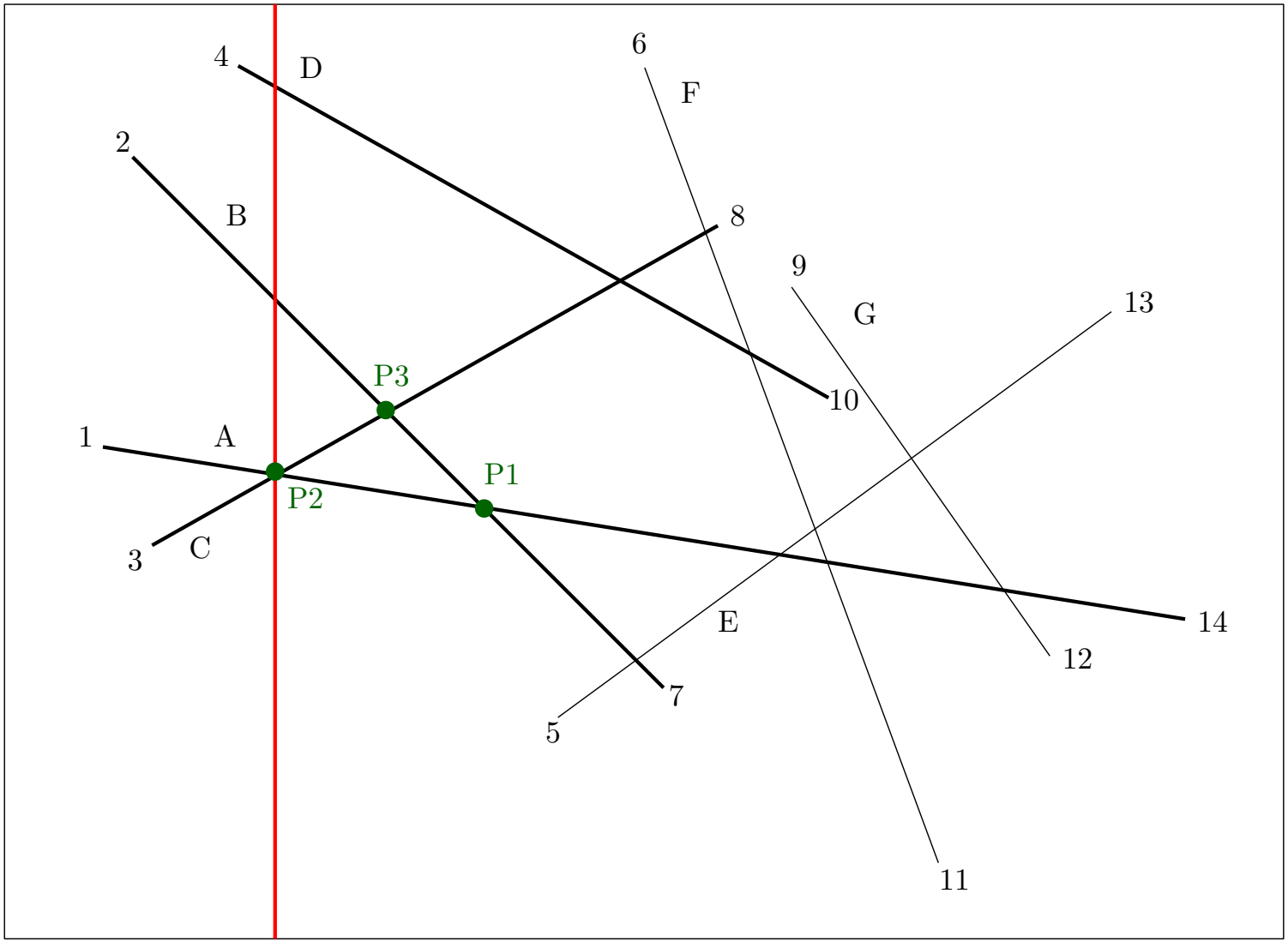
Zametací hyperrovina



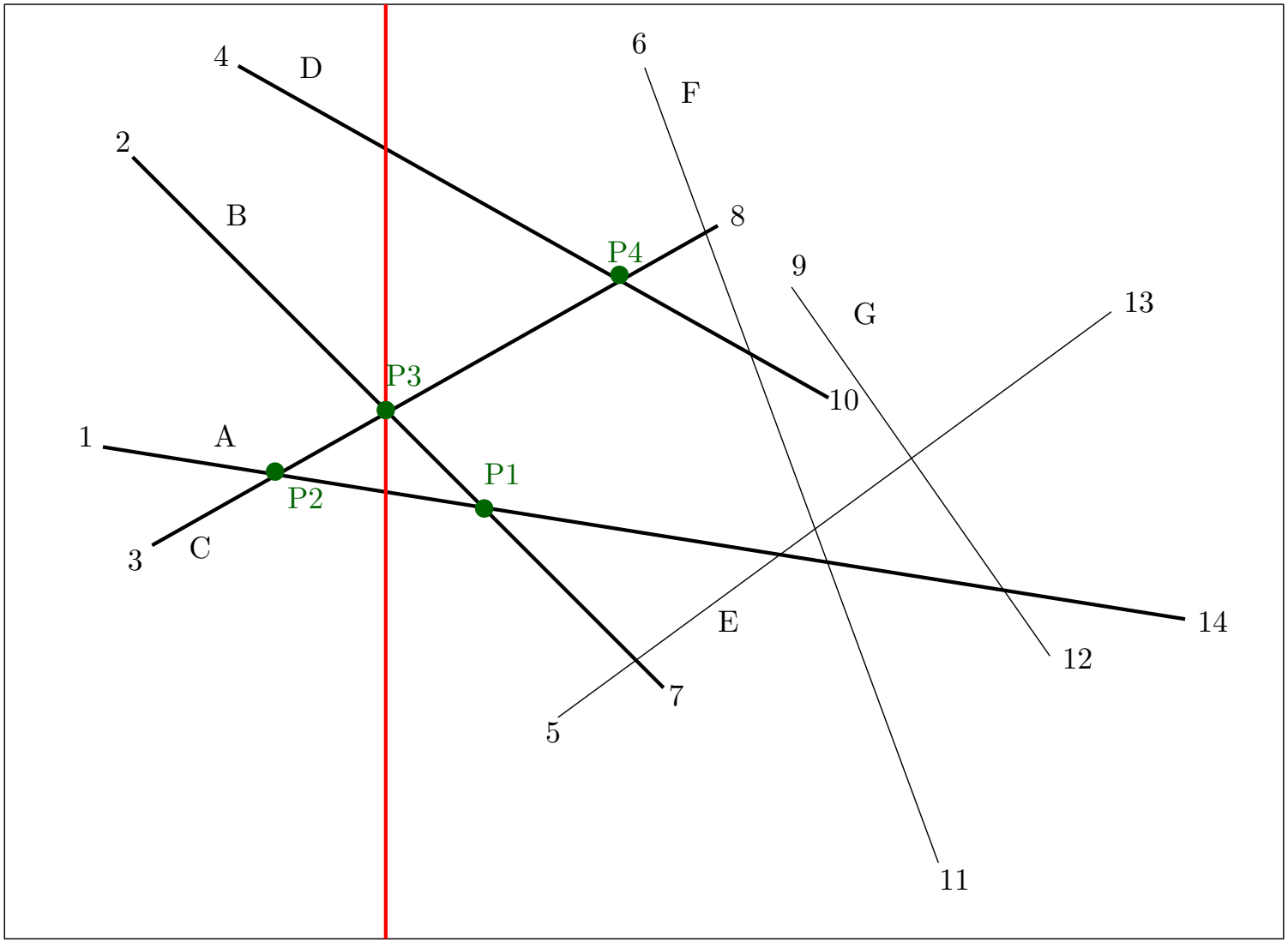
Zametací hyperrovina



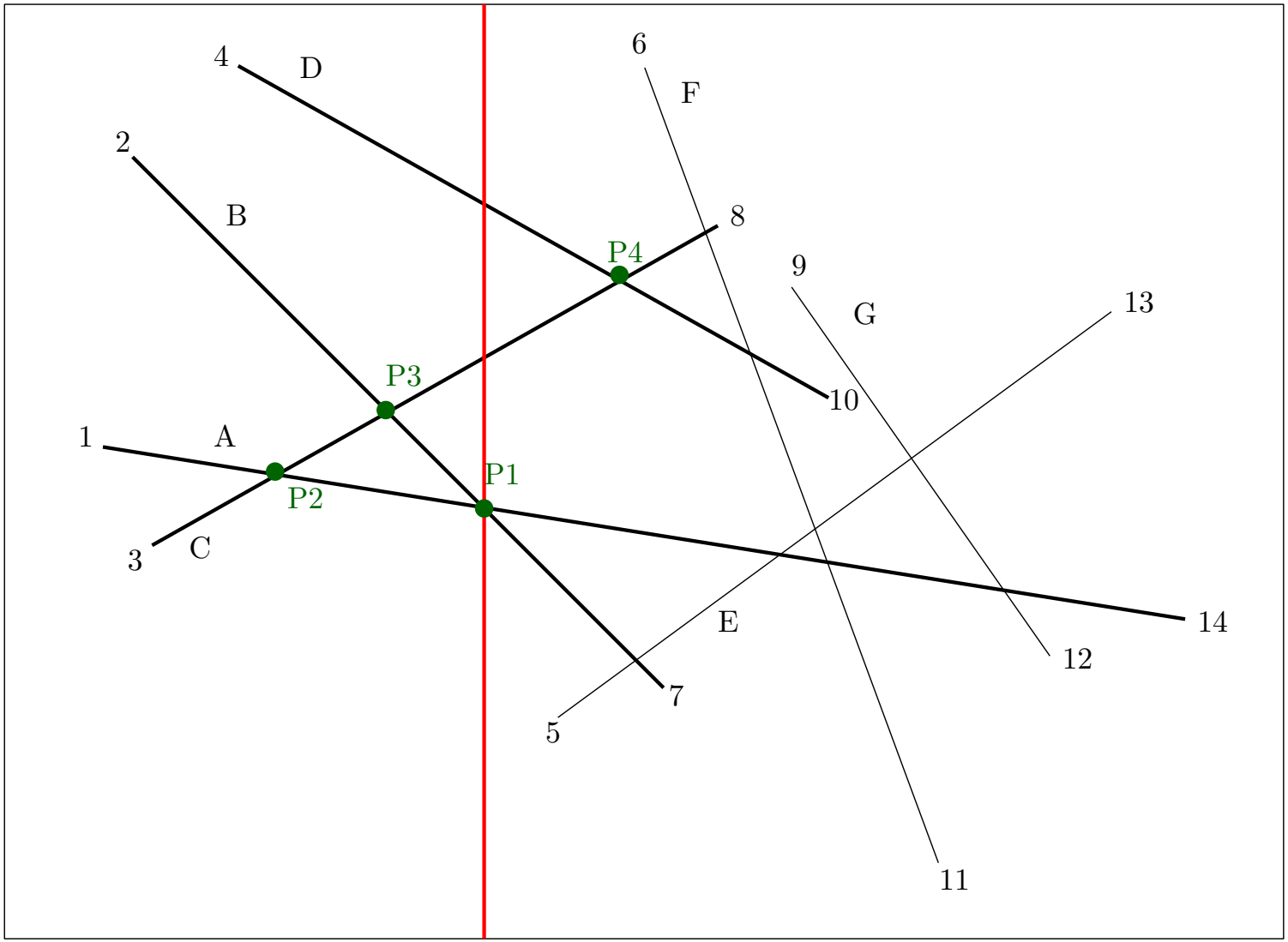
Zametací hyperrovina



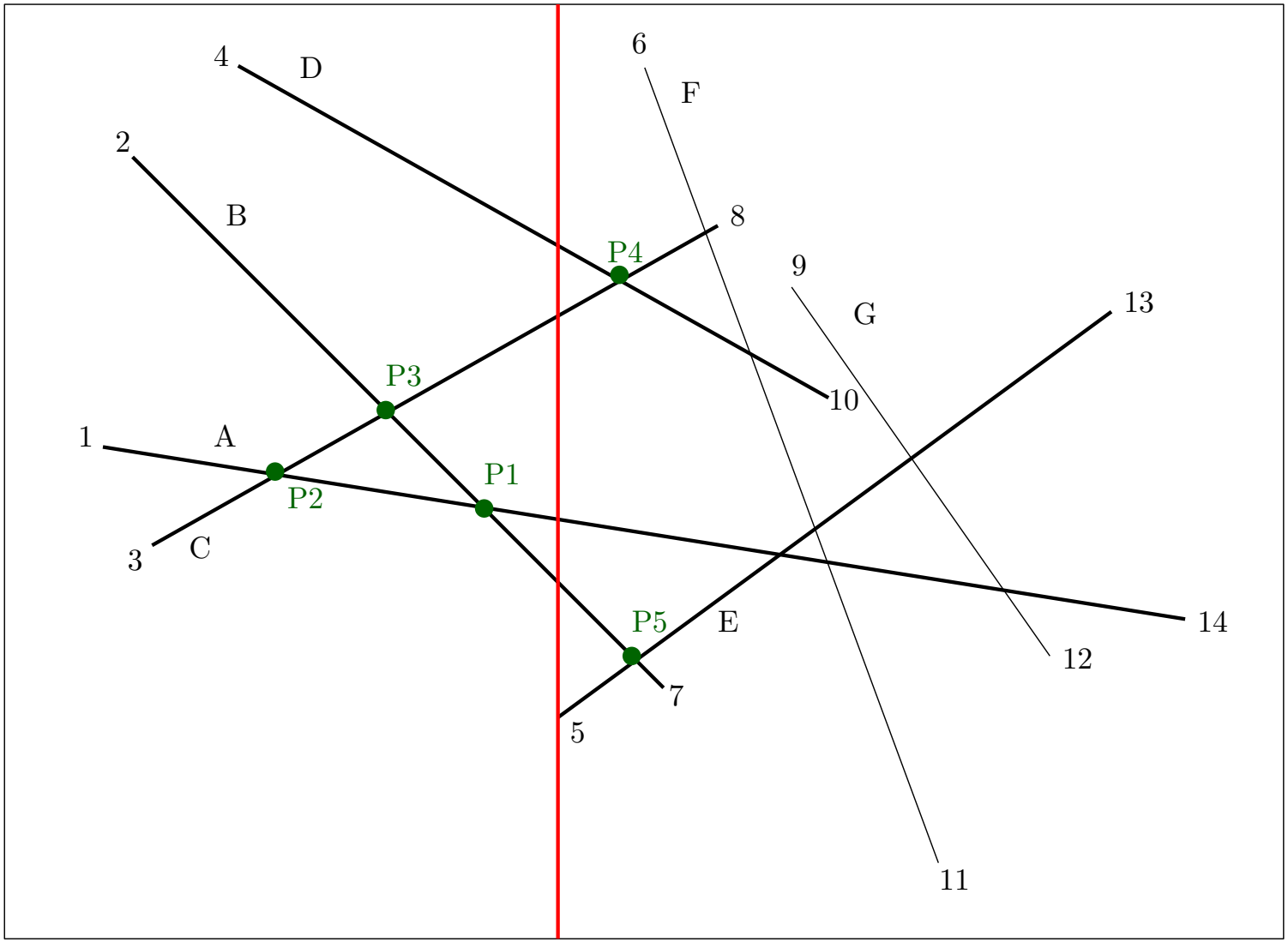
Zametací hyperrovina



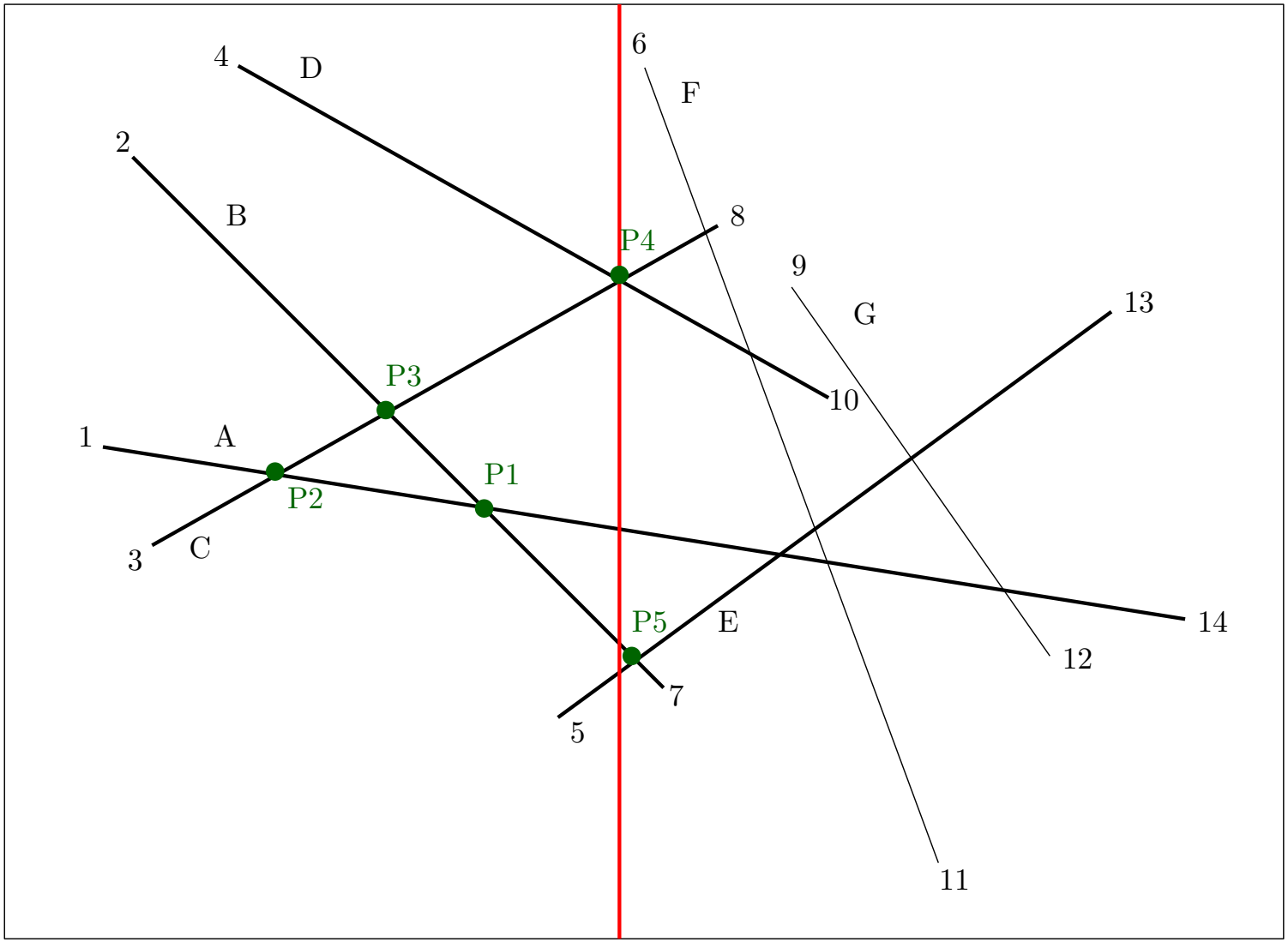
Zametací hyperrovina



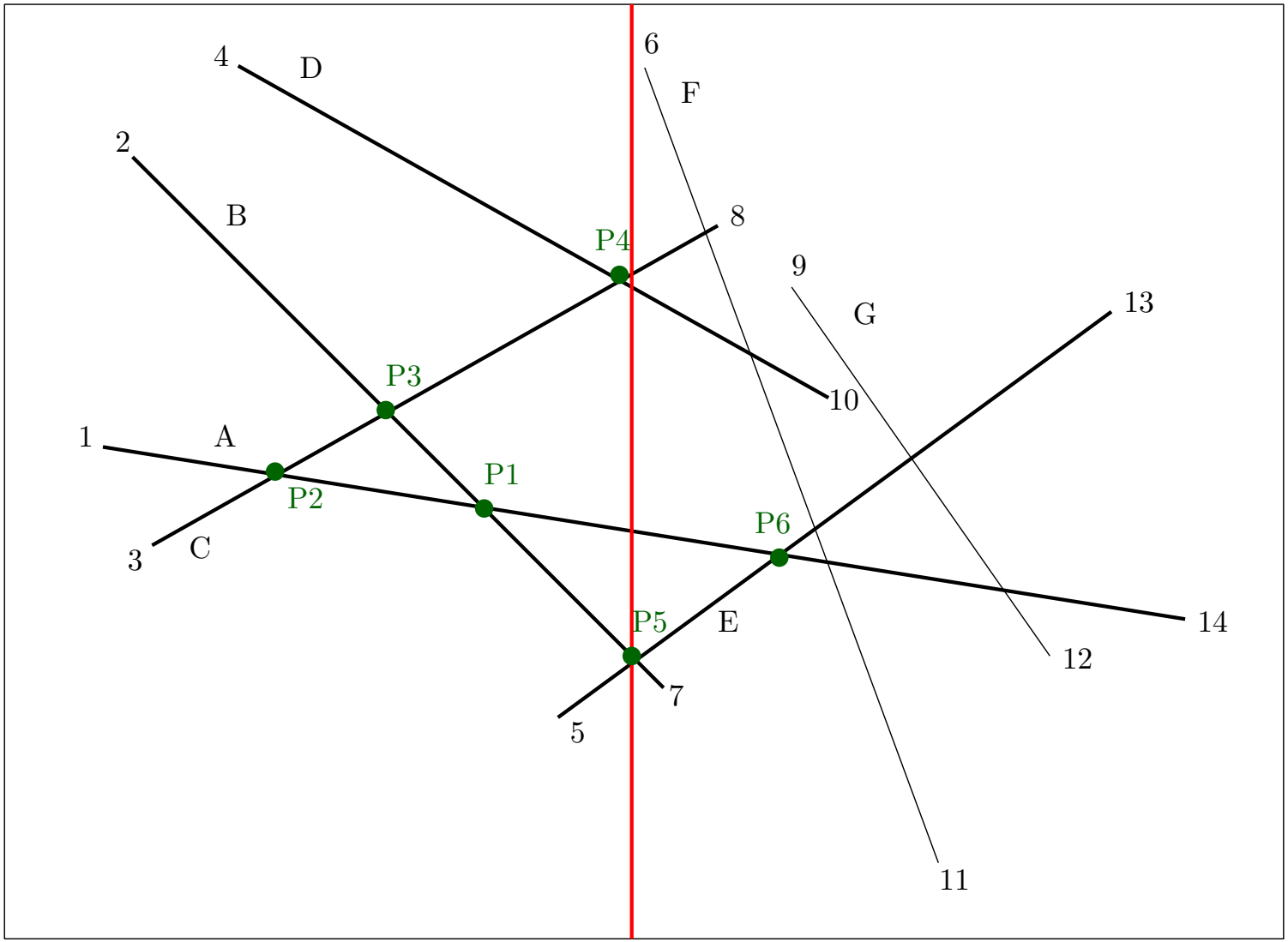
Zametací hyperrovina



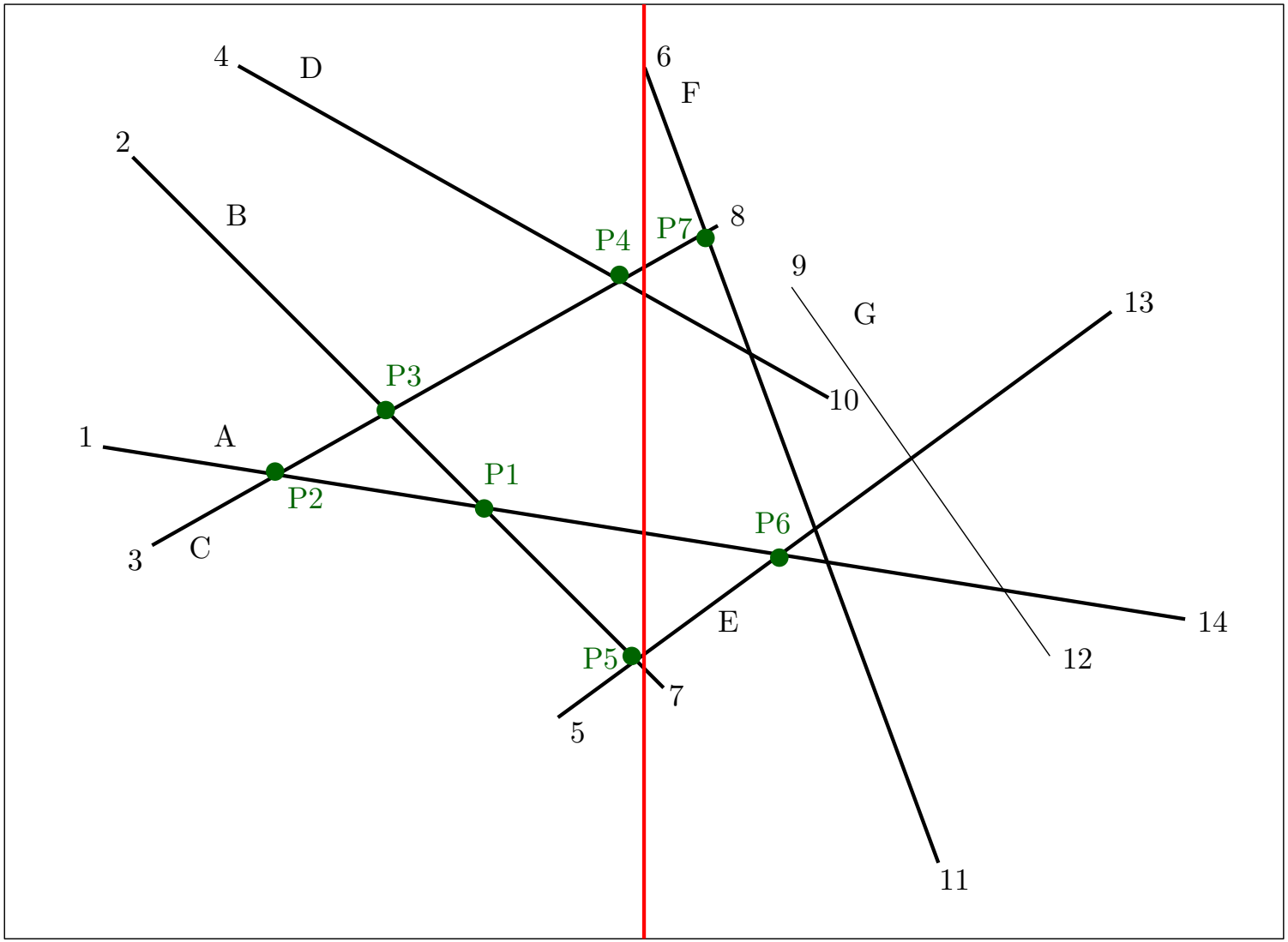
Zametací hyperrovina



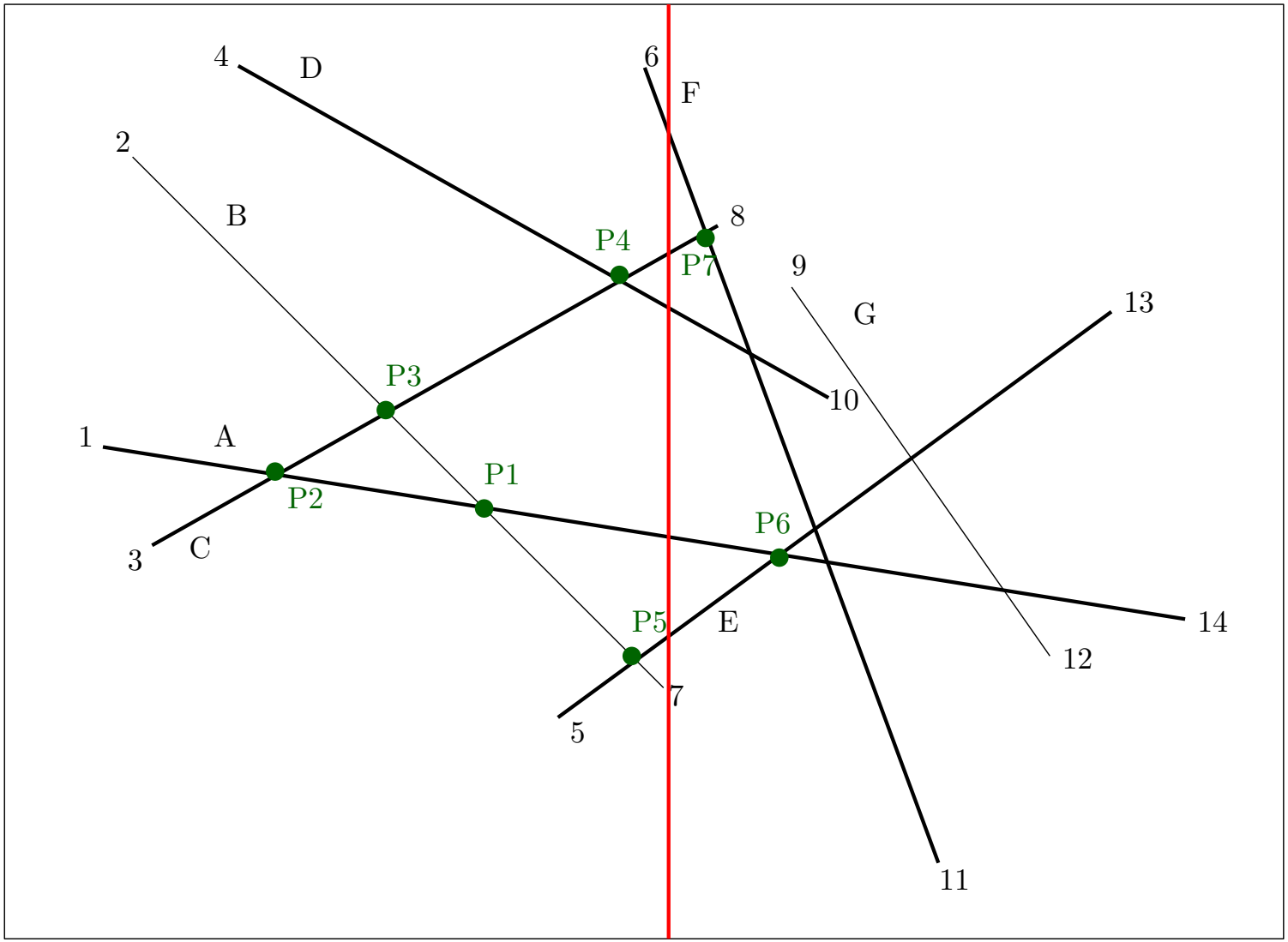
Zametací hyperrovina



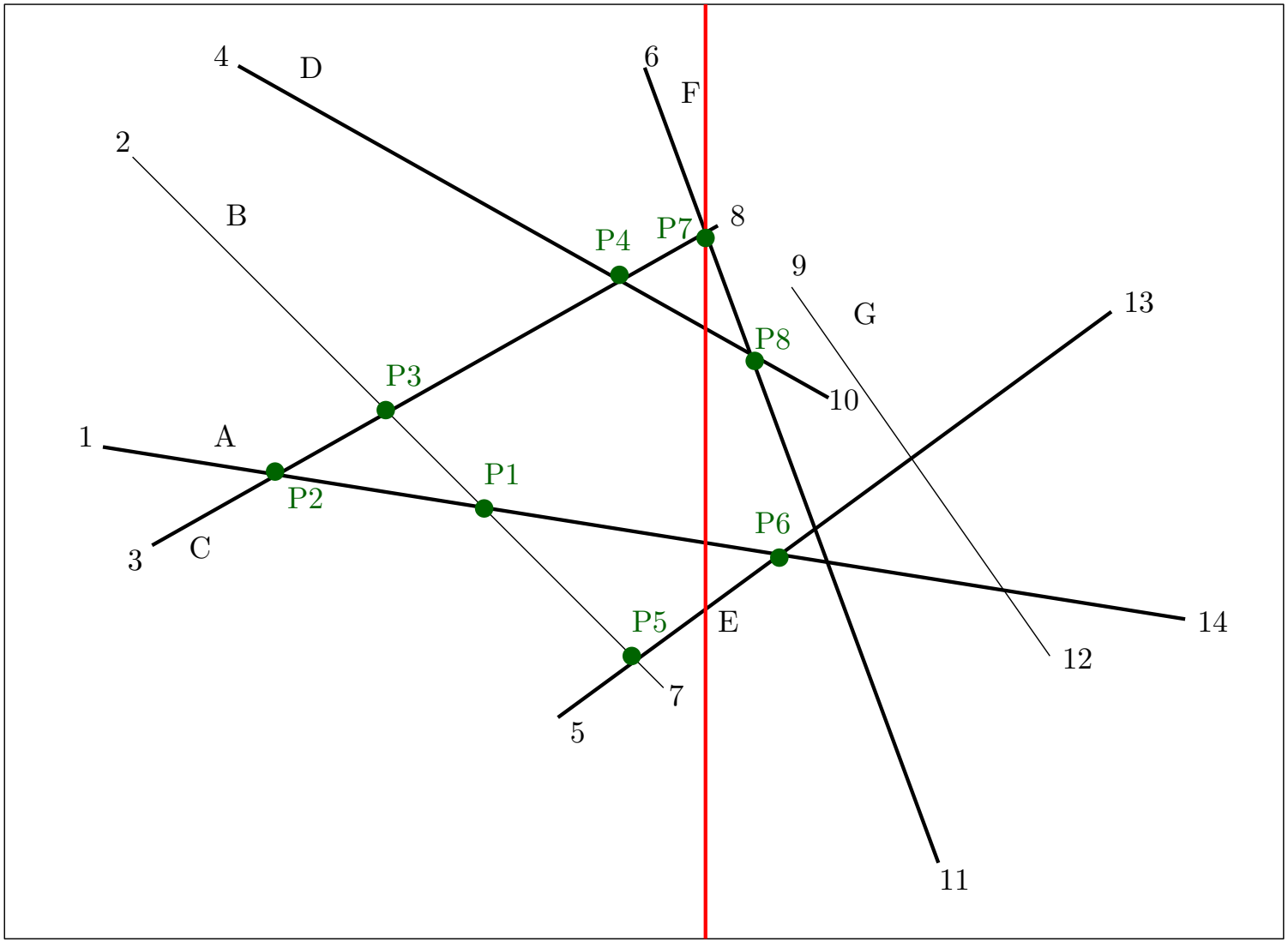
Zametací hyperrovina



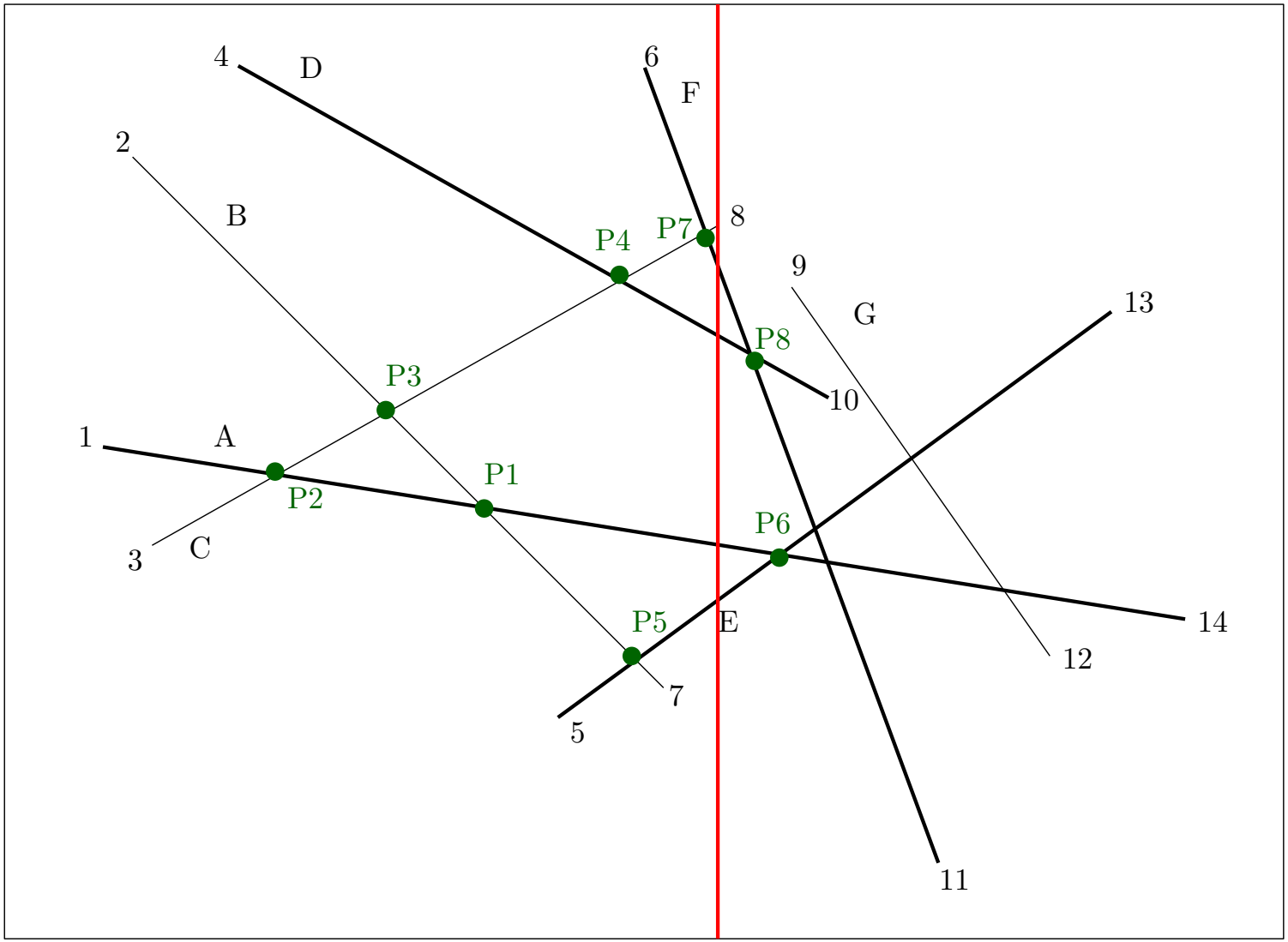
Zametací hyperrovina



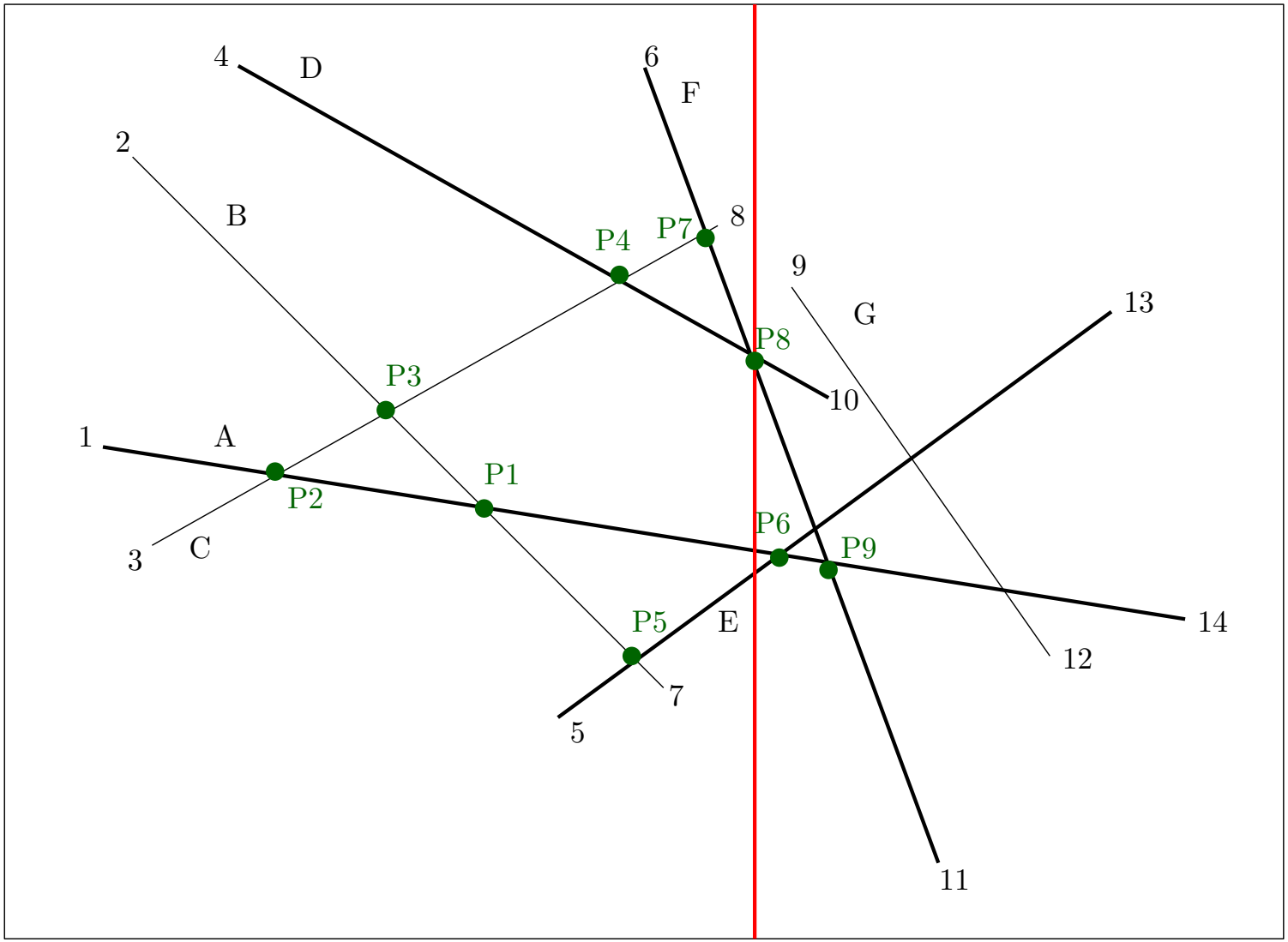
Zametací hyperrovina



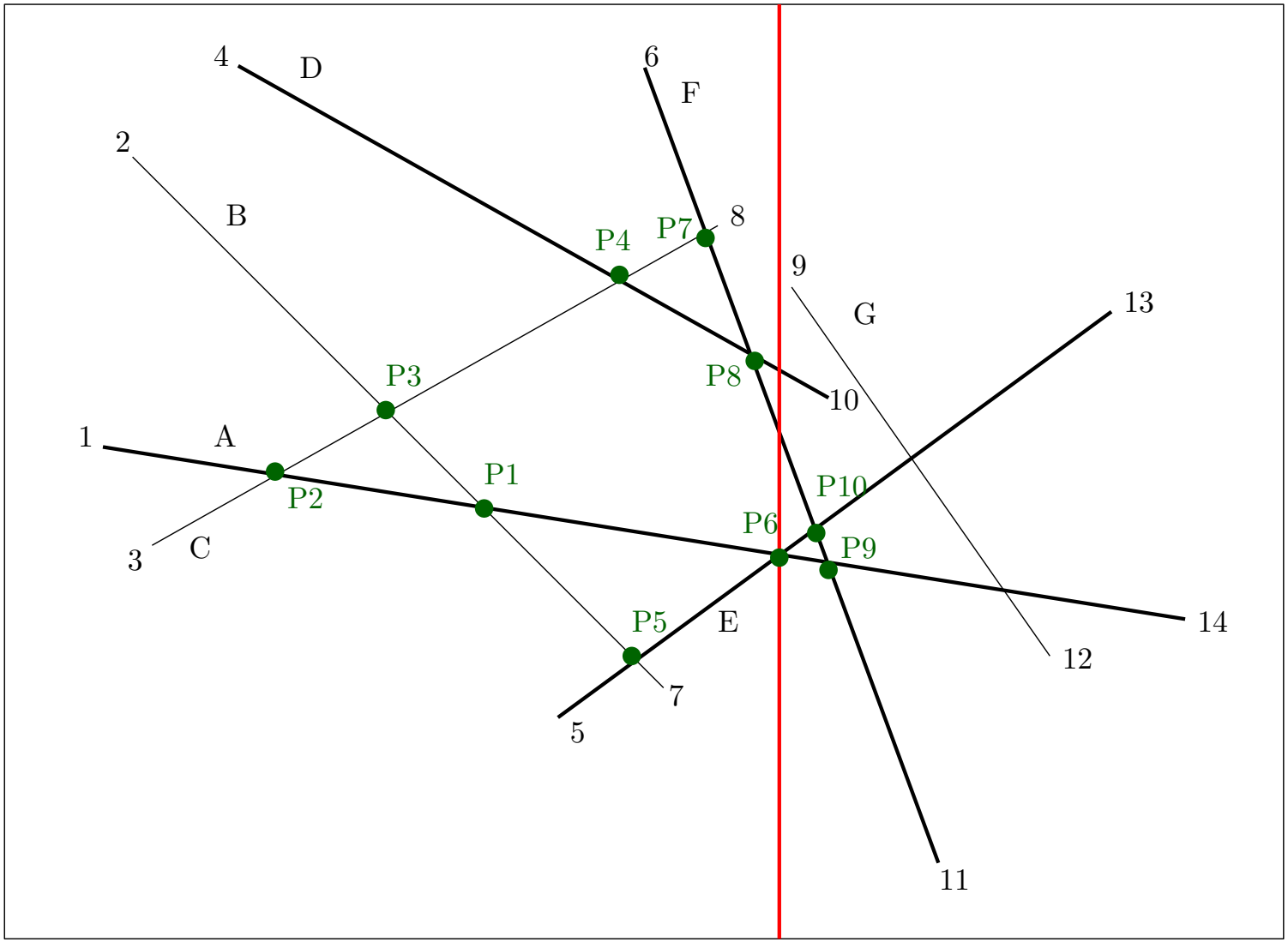
Zametací hyperrovina



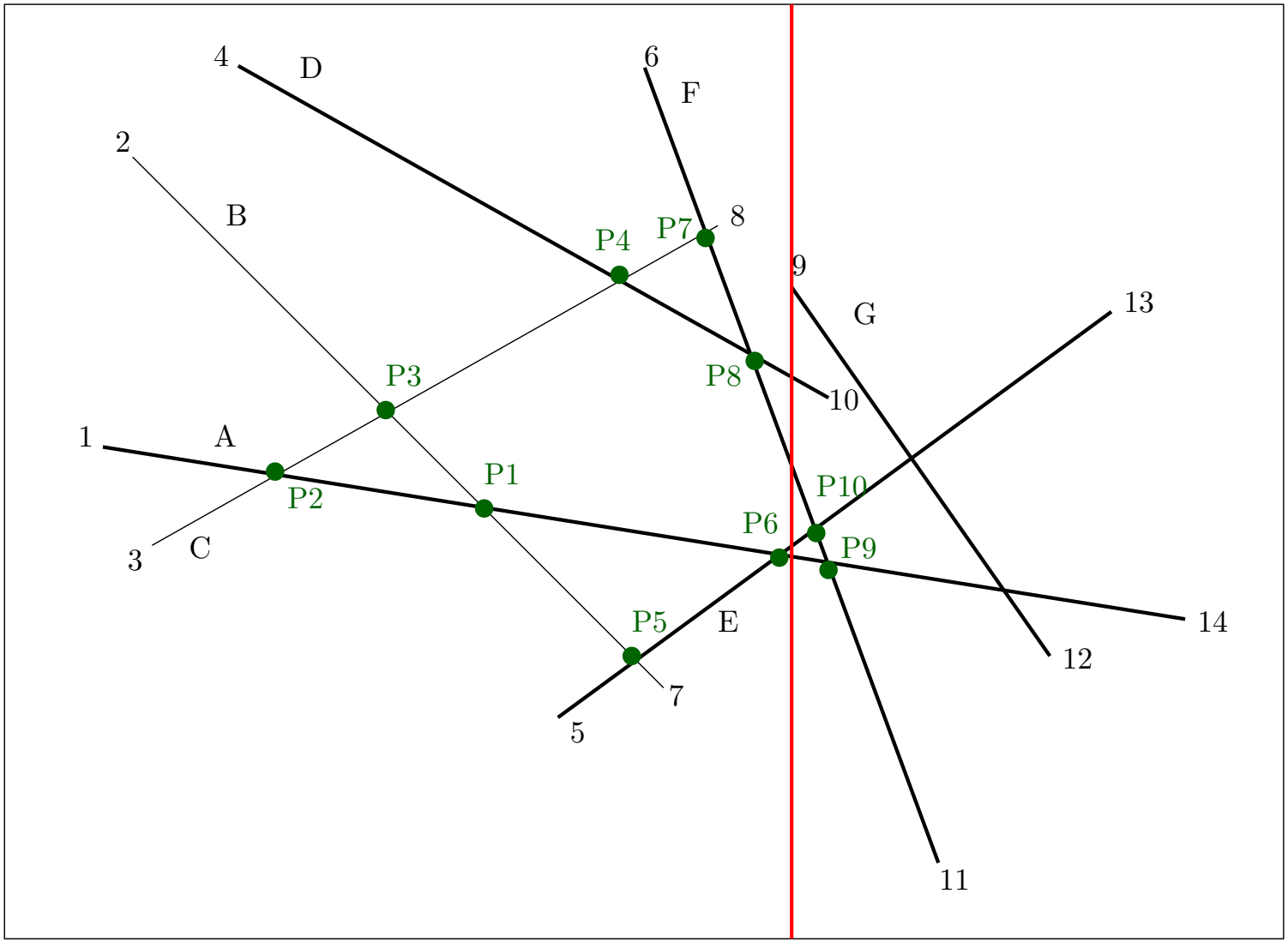
Zametací hyperrovina



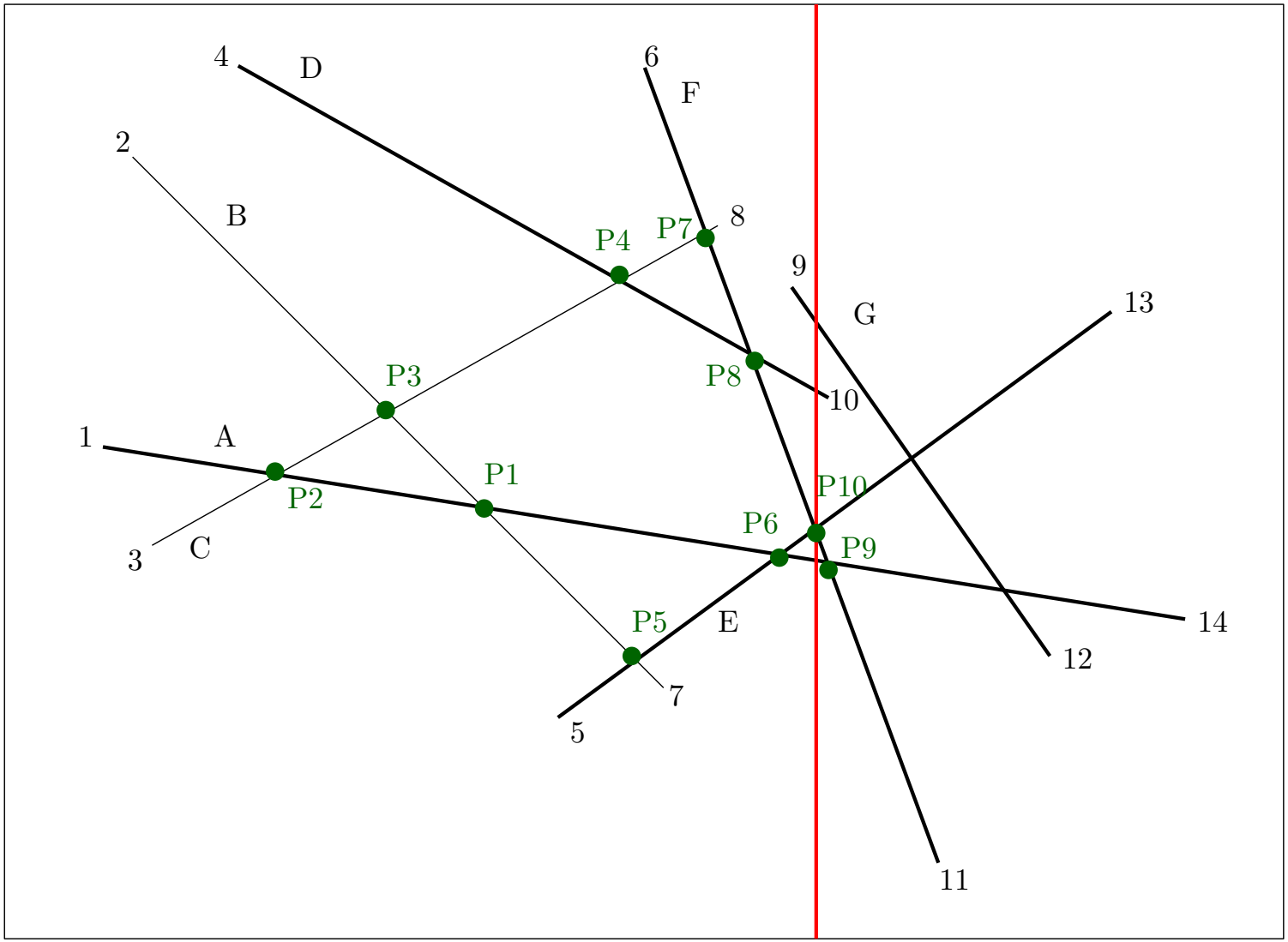
Zametací hyperrovina



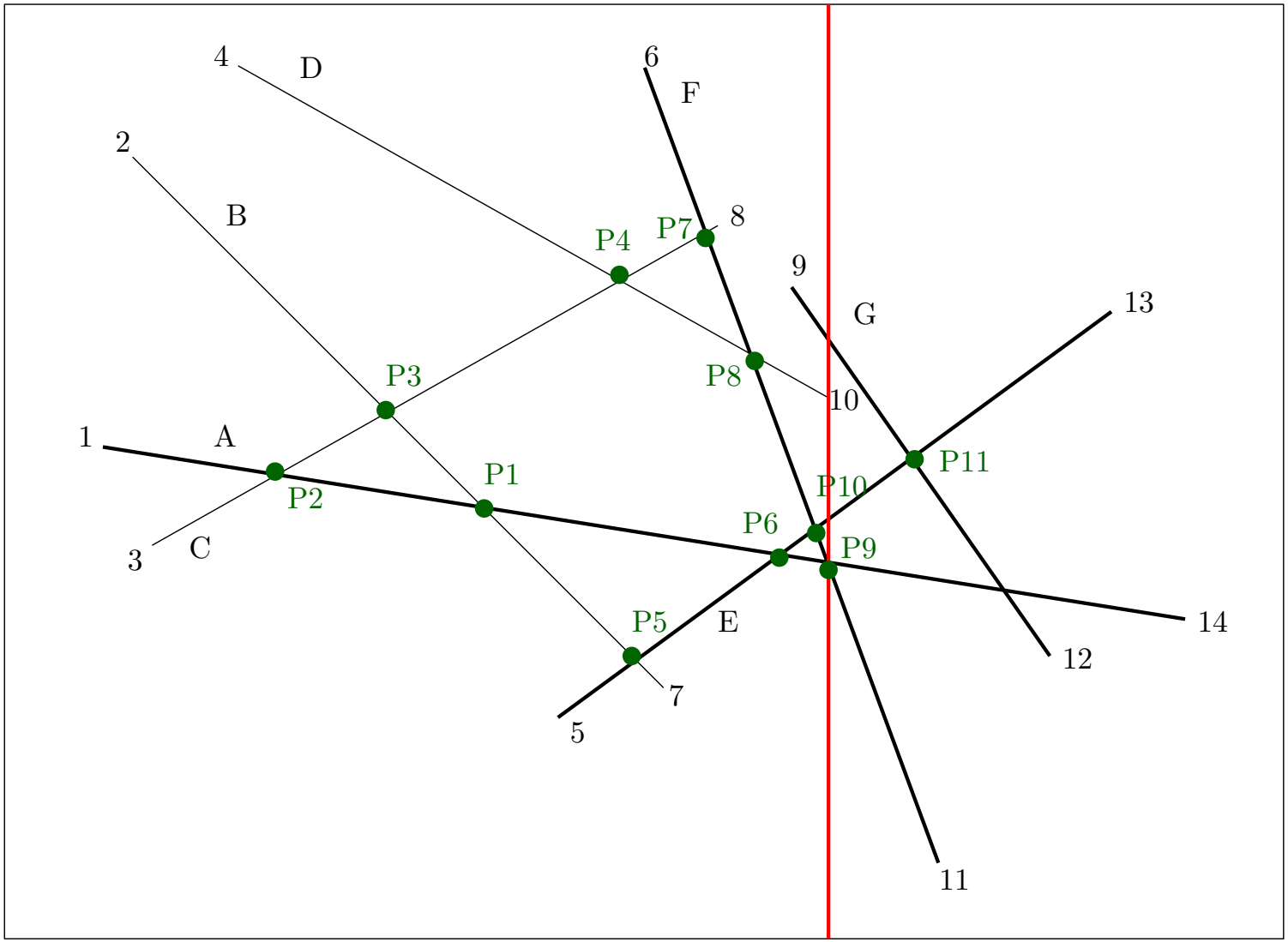
Zametací hyperrovina



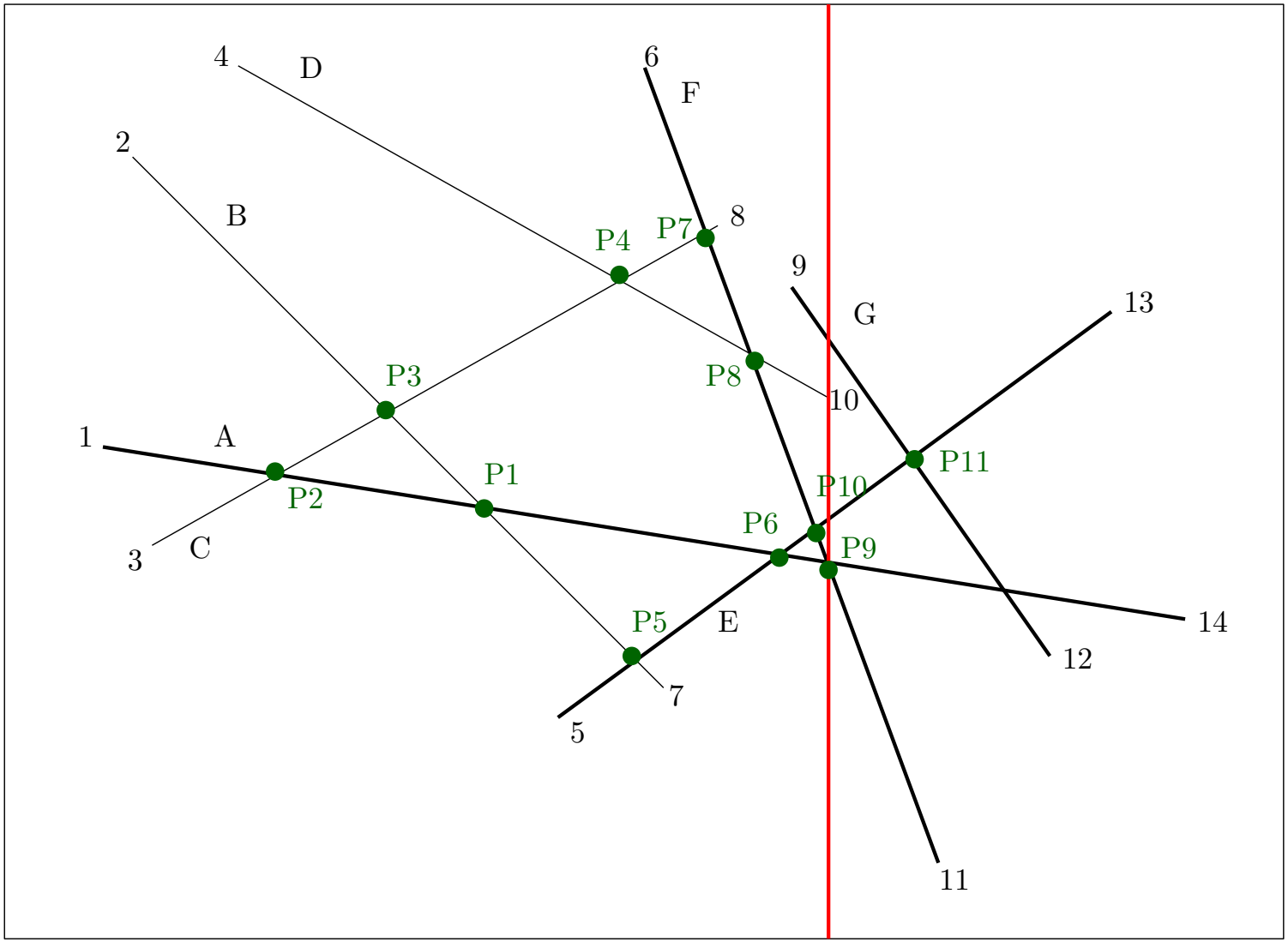
Zametací hyperrovina



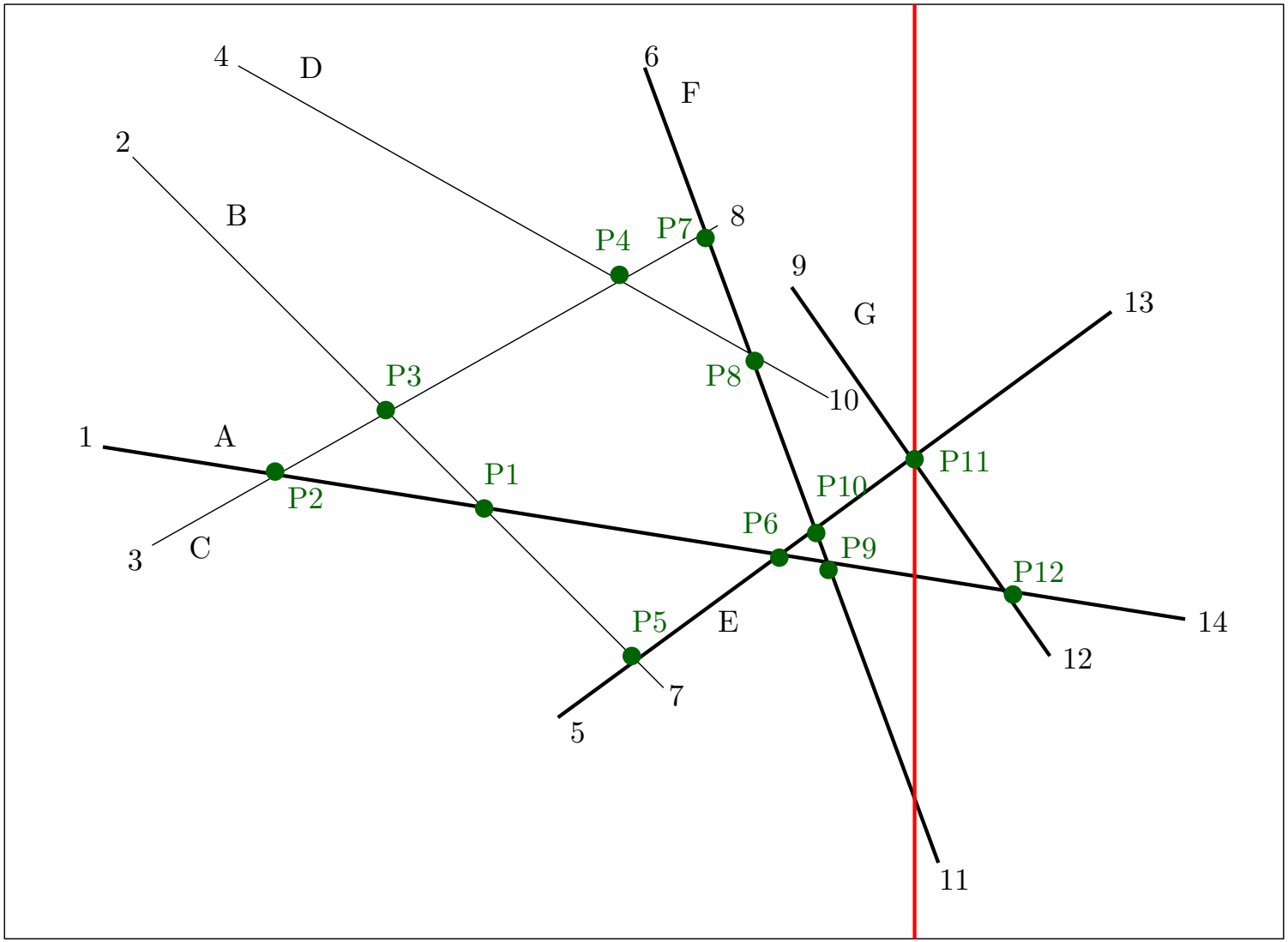
Zametací hyperrovina



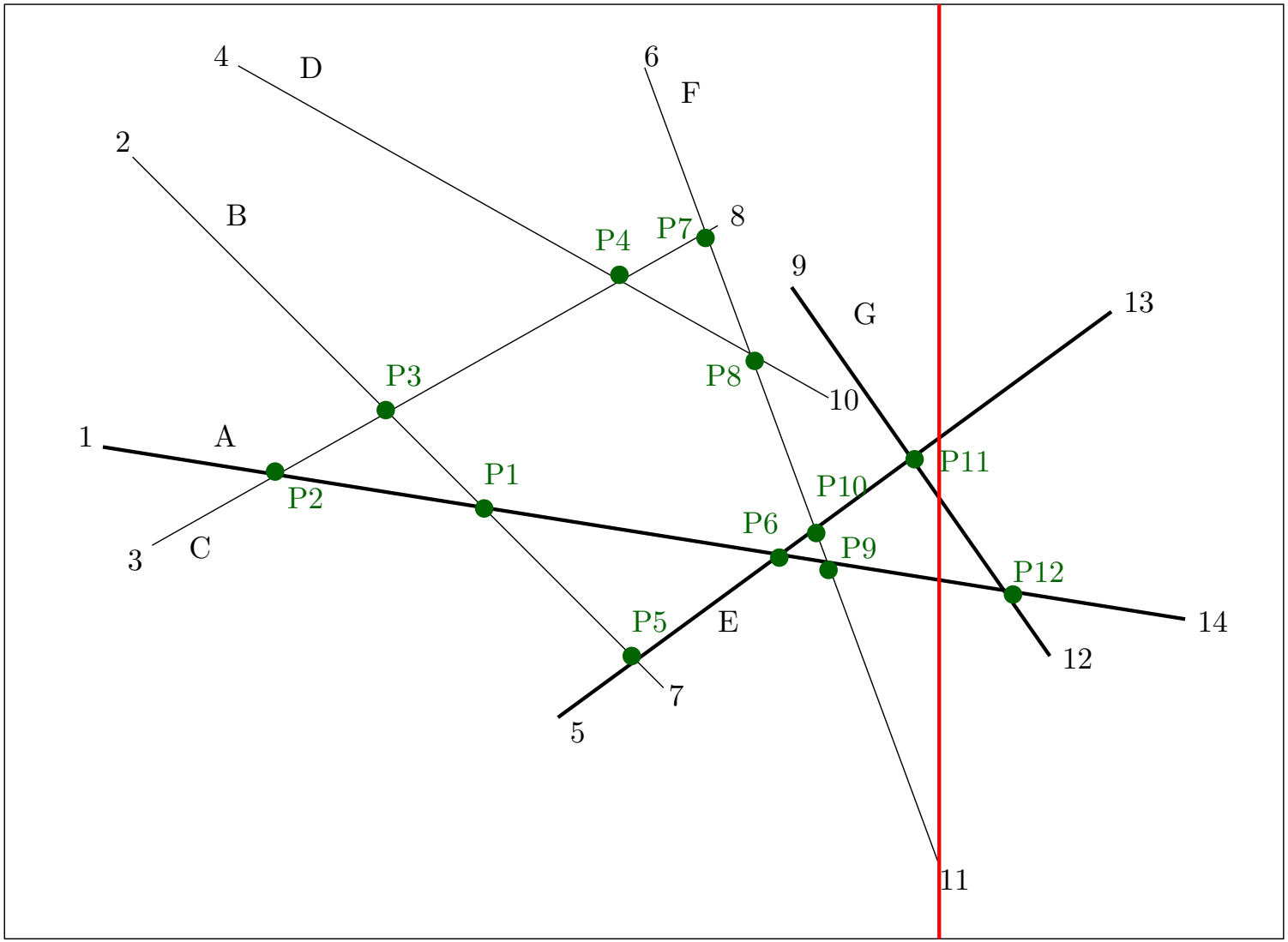
Zametací hyperrovina



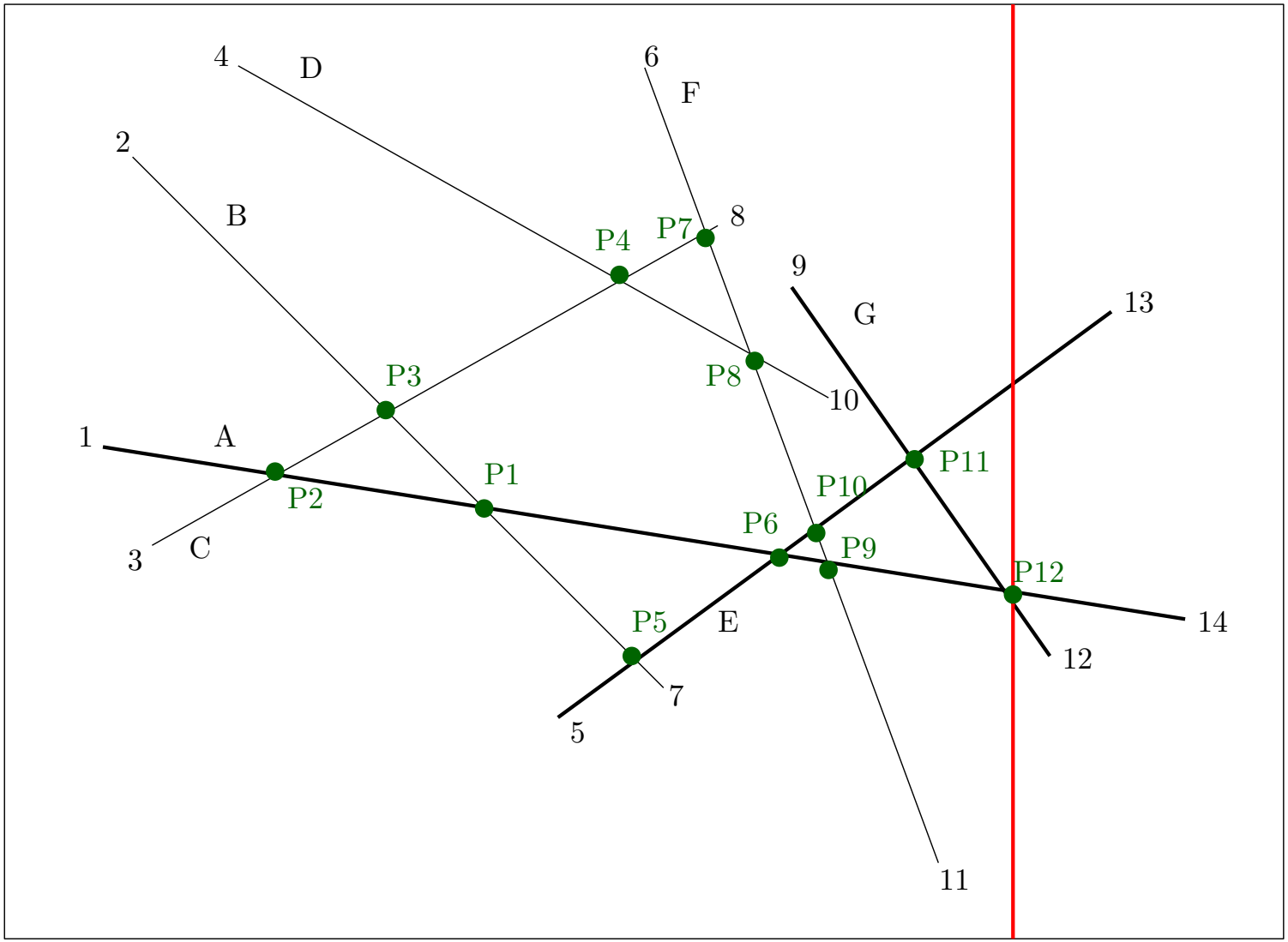
Zametací hyperrovina



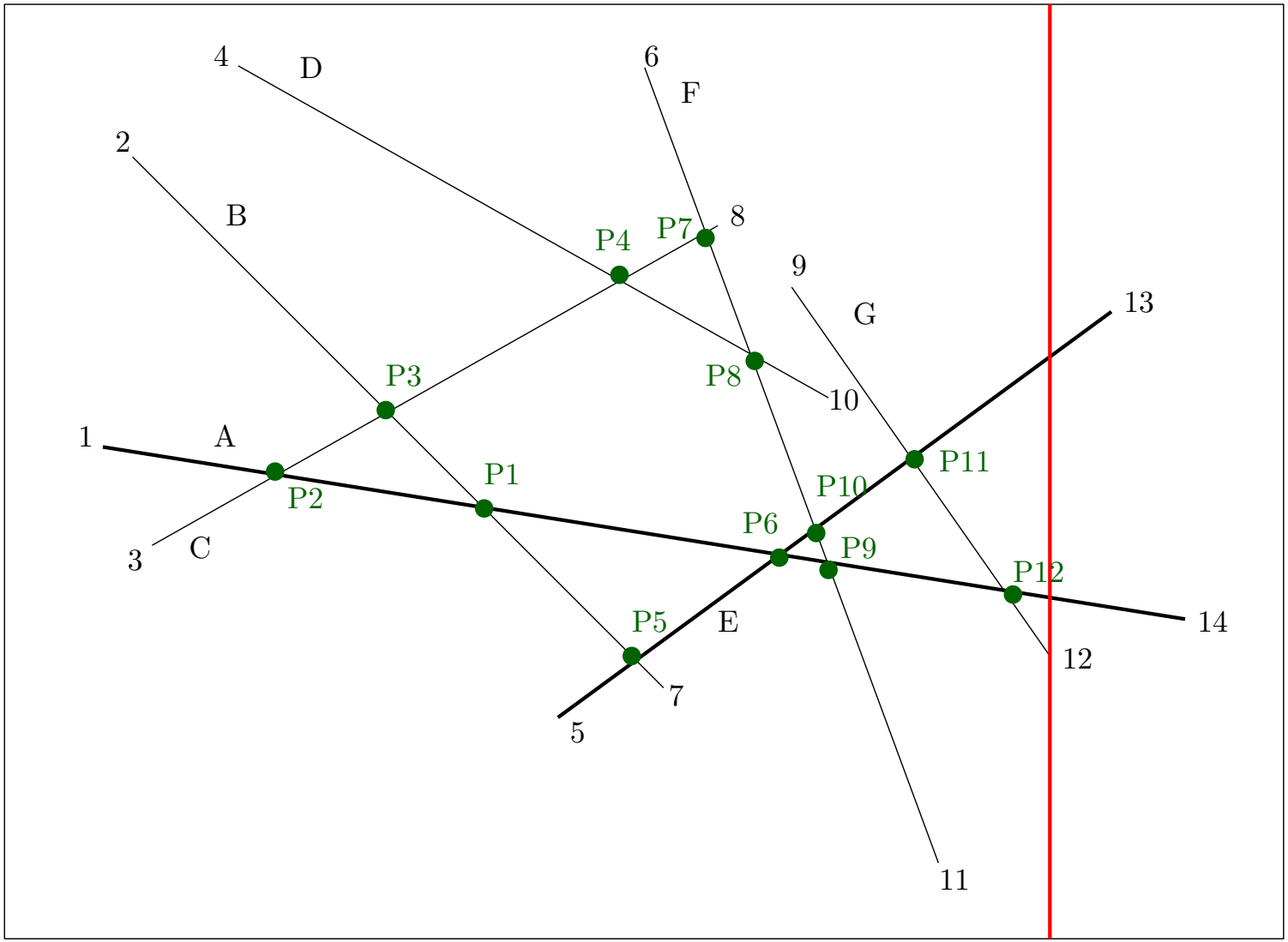
Zametací hyperrovina



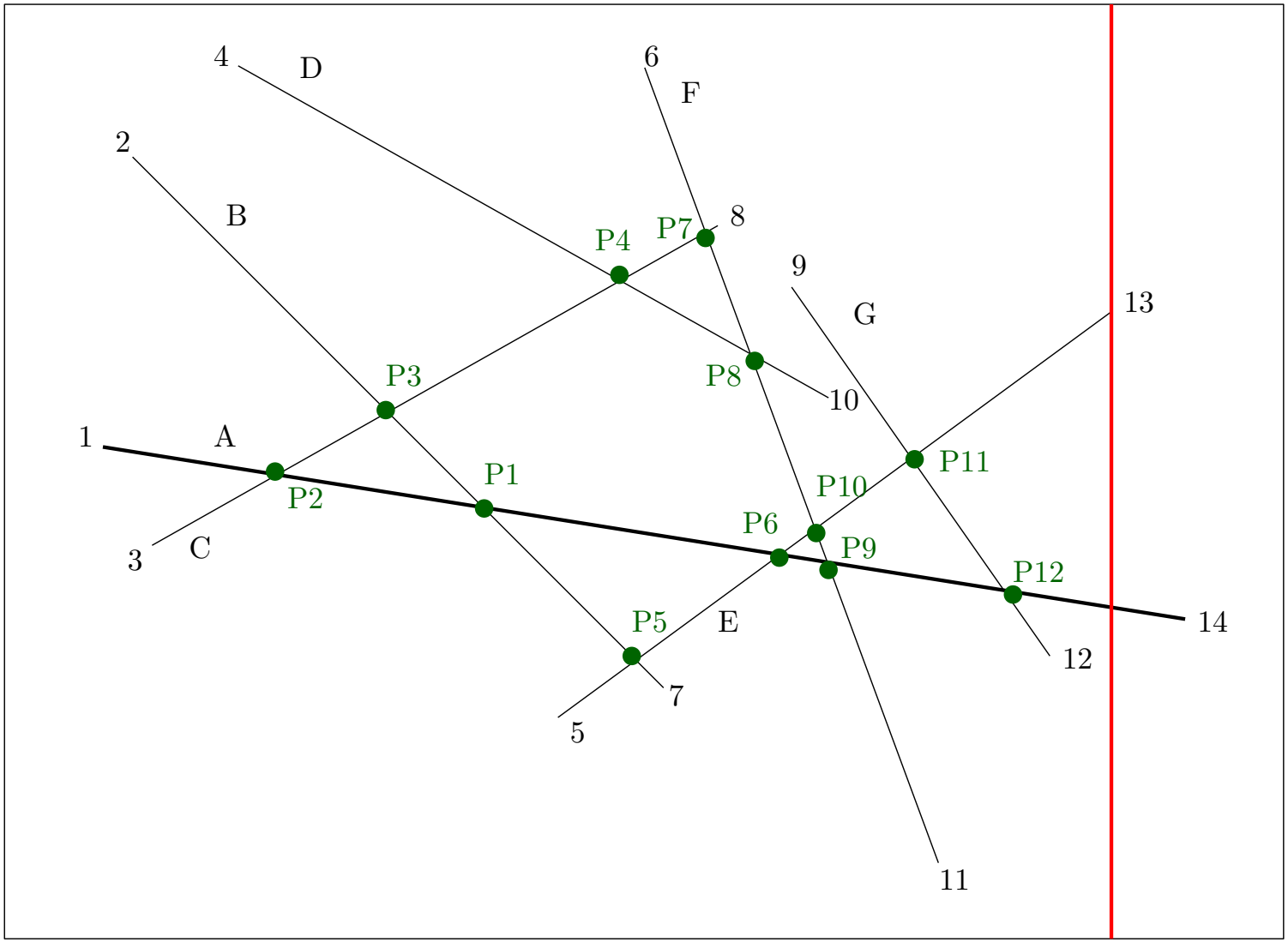
Zametací hyperrovina



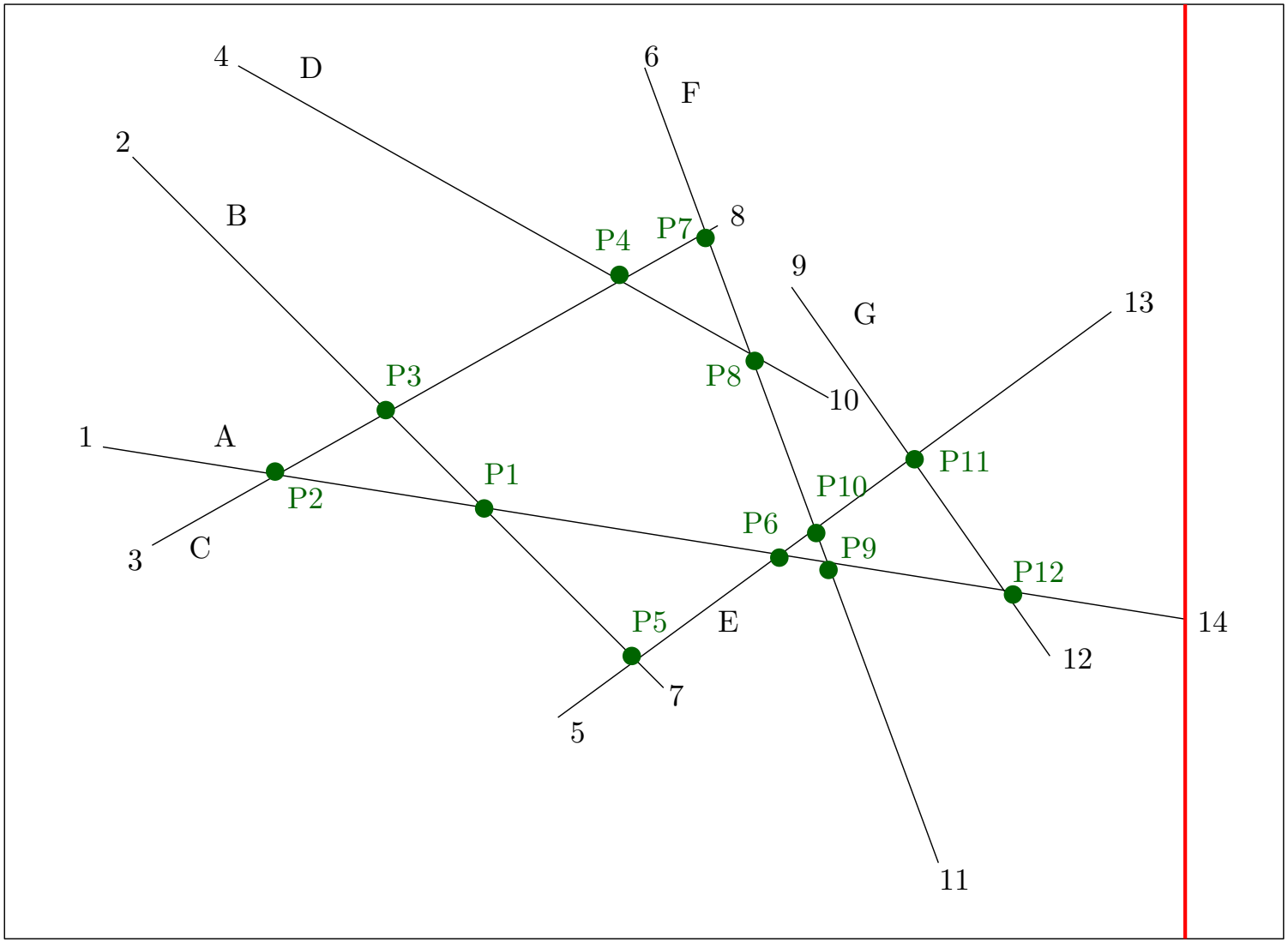
Zametací hyperrovina



Zametací hyperrovina



Zametací hyperrovina



Co bychom po dnešku měli znát

Témata probraná na dnešní přednášce:

- Kvadrantové stromy a jejich odvozeniny
- Hledání konvexní obálky
- Zametací hyperrovina