



**OI-OPPA. European Social Fund
Prague & EU: We invest in your future.**

Trojúhelníková metoda vyhledávání umístění bodu

Lenka Saidlová
saidllen@fel.cvut.cz

4.10.2012

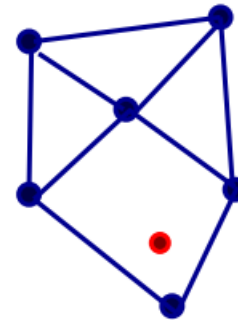


Obsah

- Úvod do problému
- Kirkpatrickova metoda
 - Triangulace
 - Postup
 - Datová struktura
 - Vyhledávání
 - Shrnutí

Poloha bodu

- Vstupy:
 - Rozdělená rovina S na několik polygonů
 - Bod P
- Výstup:
 - V jakém polygonu se bod P nachází



Metody

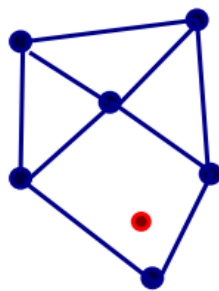
- **NE:** procházet každý polygon zvlášť
- **Minulý týden:**
 - Metoda pásů (slabs)
 - Metoda monotónního rozdělení roviny
 - Lichoběžníková (Trapezoidal) mapa
- **Dnes:**
 - Trojúhelníková metoda

Trojúhelníková (Kirkpatrickova) metoda

- Časová složitost vyhledávání $O(\log n)$
- Paměťová složitost $O(n)$
- Principem je rozdělit polygony na trojúhelníky a následně vytvořit acyklický orientovaný graf (DAG)
- **Cíl:** nalézt trojúhelník, ve kterém se bod nachází

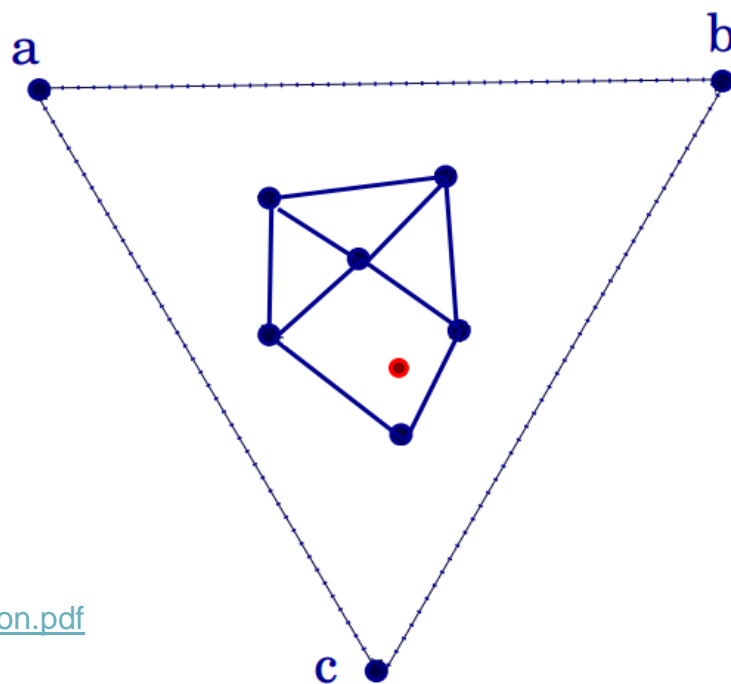
Předpoklady

- Daná rovinná oblast je trojúhelníková
- Pokud ne, vytvoříme konvexní obálku ve tvaru trojúhelníku



Obrázek převzat z:

<http://www.cs.ucsb.edu/~suri/cs235/Location.pdf>



Triangulace

- Rozdělení polygonů na trojúhelníky
= triangulace
- Mnoho algoritmů – $O(n \log n)$ až $O(n^3)$
- Zohlednění tvaru, nekonvexní oblasti
- Greedy, Minimum Weight,....

Postup

- Odstraníme některé vrcholy T_i
- To způsobí díry, které opět rozdělíme na trojúhelníky
- Odstraňujeme vždy konstantní počet vrcholů, abychom docílili $O(1)$ času triangulace T_{i+1}
- Vrcholy odstraňujeme, dokud v T_k nevznikne pouze jeden trojúhelník

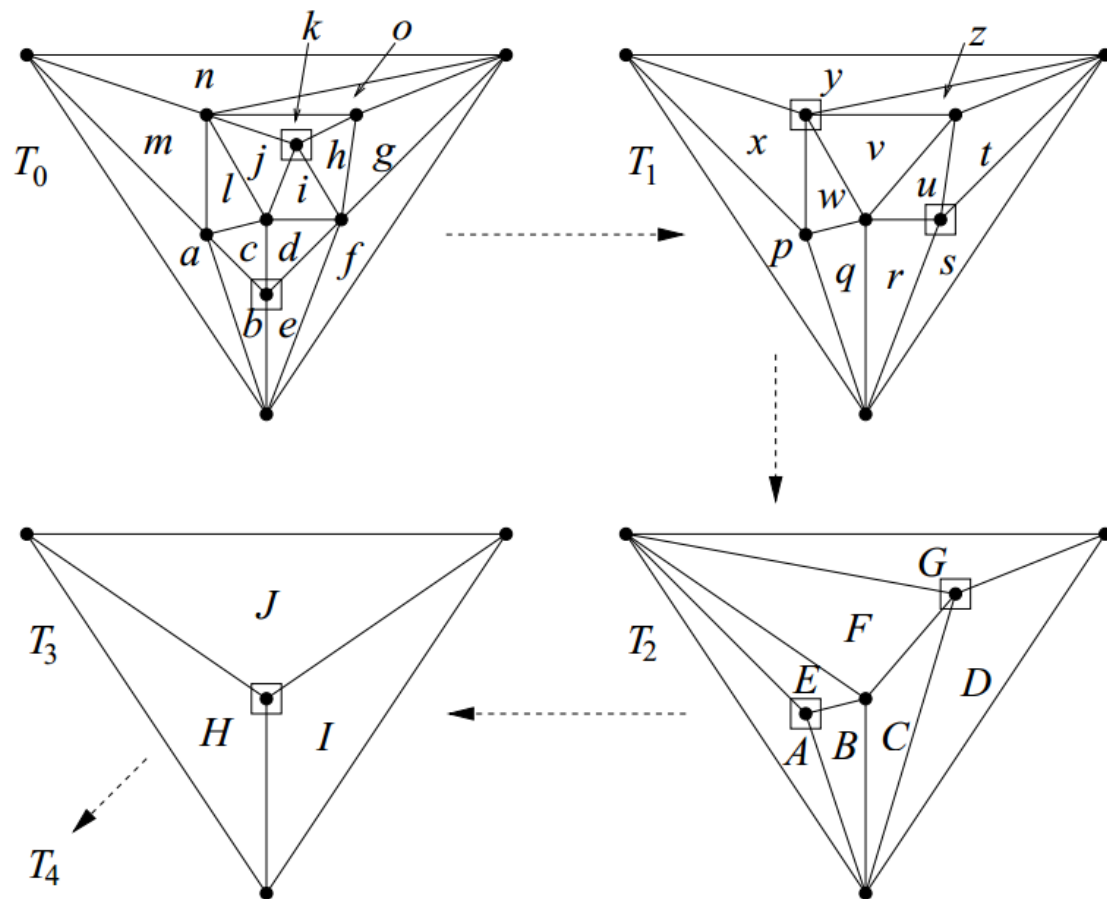
Vytvoření sekvence

- Vytvořit sekvenci triangulací T_0, T_1, \dots, T_k s těmito vlastnostmi:
 - T_0 je počáteční triangulace, T_k je vnější trojúhelník abc
 - Trojúhelníky v T_{i+1} překryjí v konstantním čase trojúhelníky z T_i
 - $k = \log n$

Které vrcholy odstranit?

- Vnitřní
- Žádné 2 vrcholy, které odstraňujeme, nesmí být vedle sebe → jednodušší triangulace, nezávisle na sobě
- S maximálním stupněm 8
- Odstraňujeme maximální počet vrcholů, které splňují předchozí podmínky

Názorný příklad

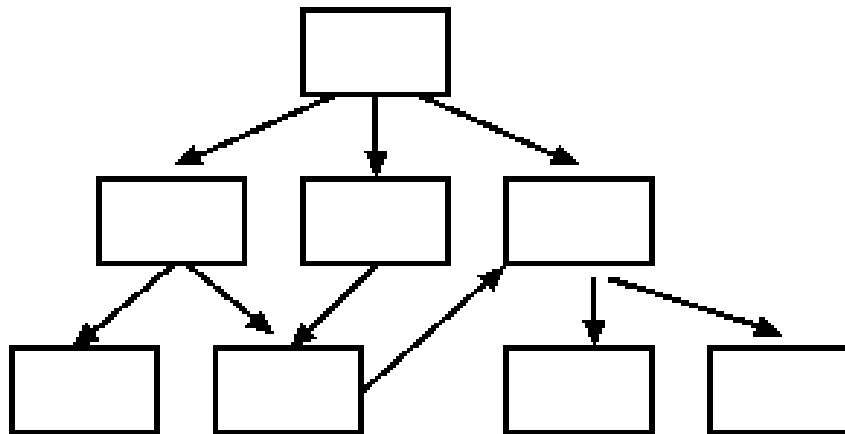


Obrázek převzat z:

<http://www.cs.ucsb.edu/~suri/cs235/Location.pdf>

Datová struktura

- DAG – Directed acyclic graph
= orientovaný acyklický graf



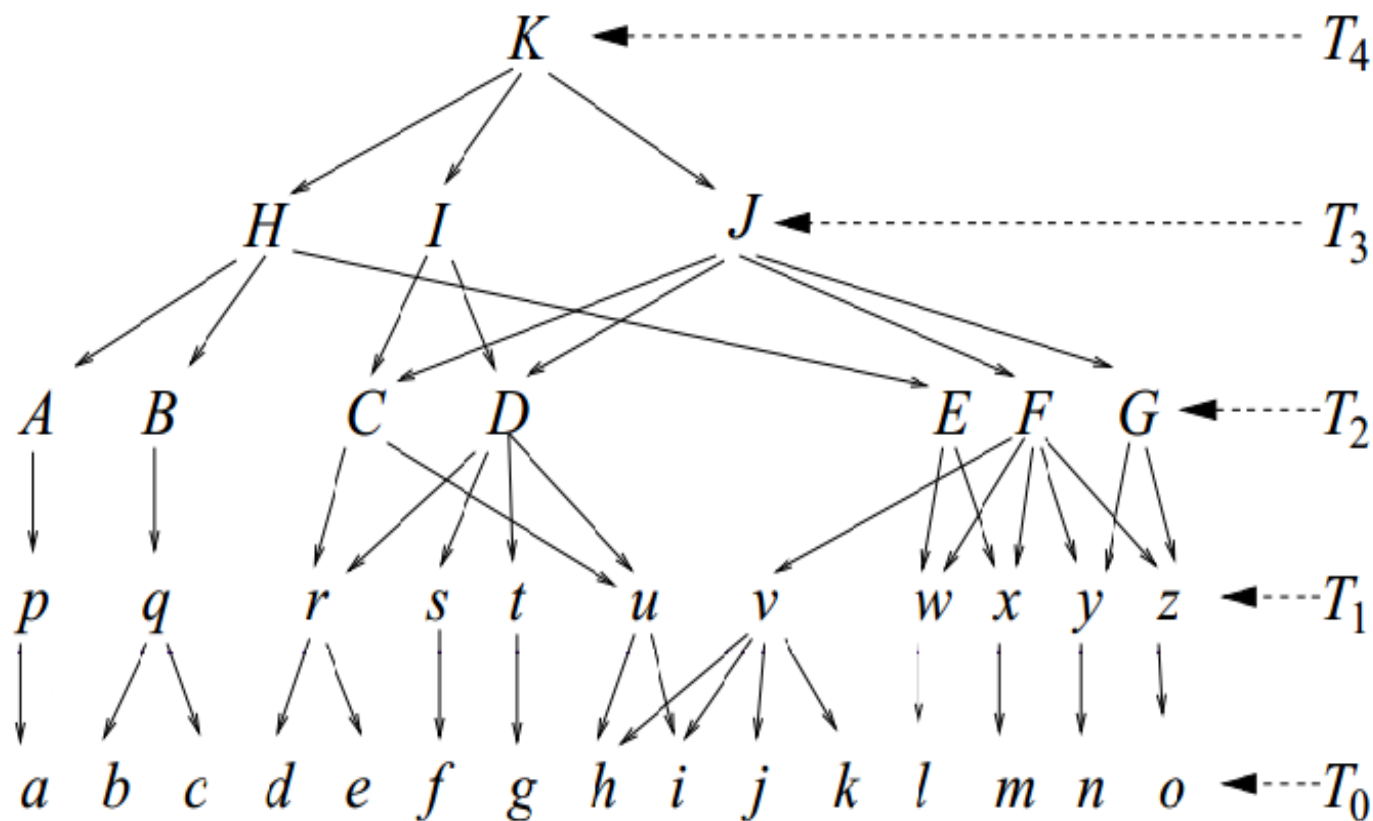
Obrázek převzat z:

<http://home.cc.umanitoba.ca/~frist/PLNT7690/lec11/lec11.3.html>

Datová struktura DAG

- Kořen odpovídá triangulaci T_k – pouze 1 ▲
- Potomci kořene odpovídají triangulaci T_{k-1}
- Každý uzel triangulace T_{i+1} má ukazetele na všechny trojúhelníky triangulace T_i , které překrývá
- Každý uzel i -té úrovně odpovídá trojúhelníku triangulace T_{k-i}

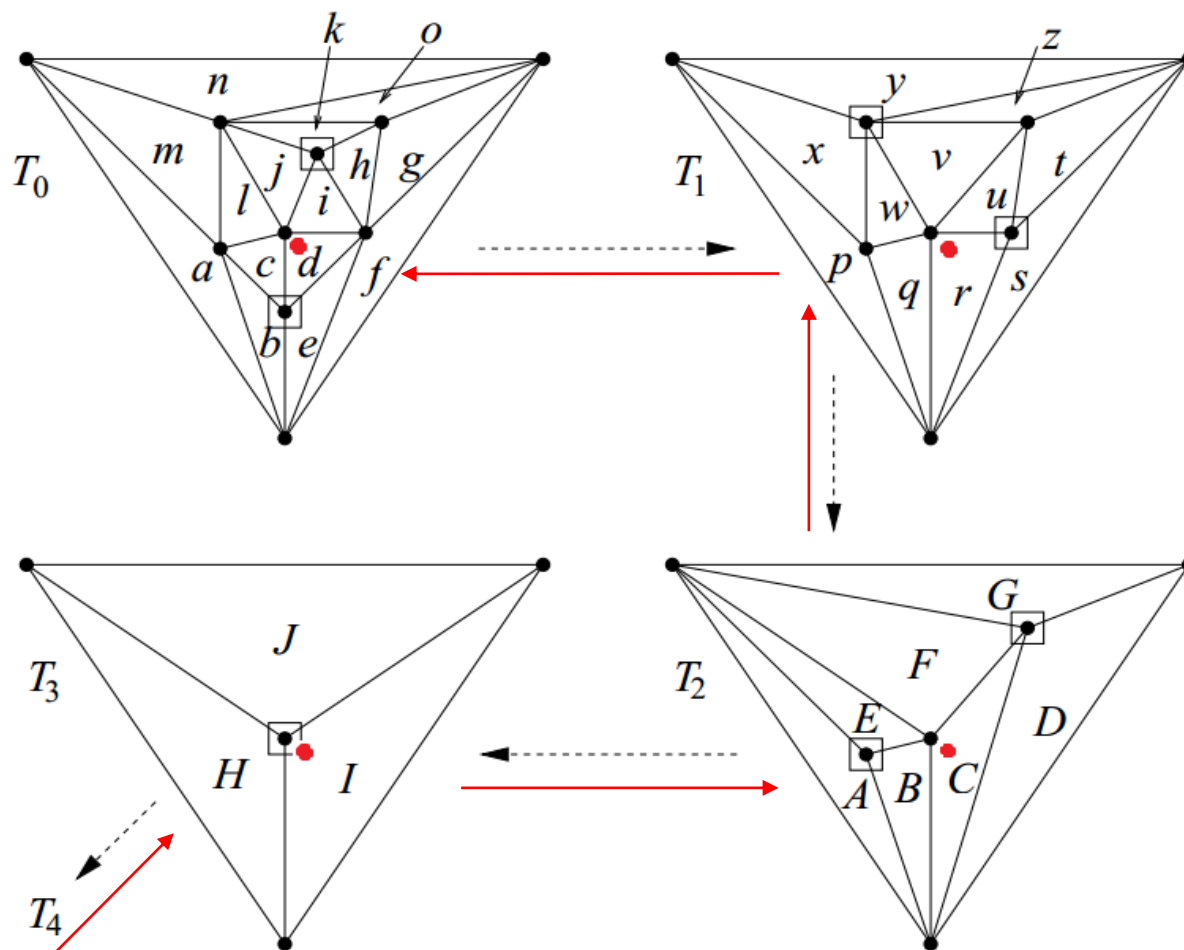
DAG názorného příkladu



Obrázek převzat z:

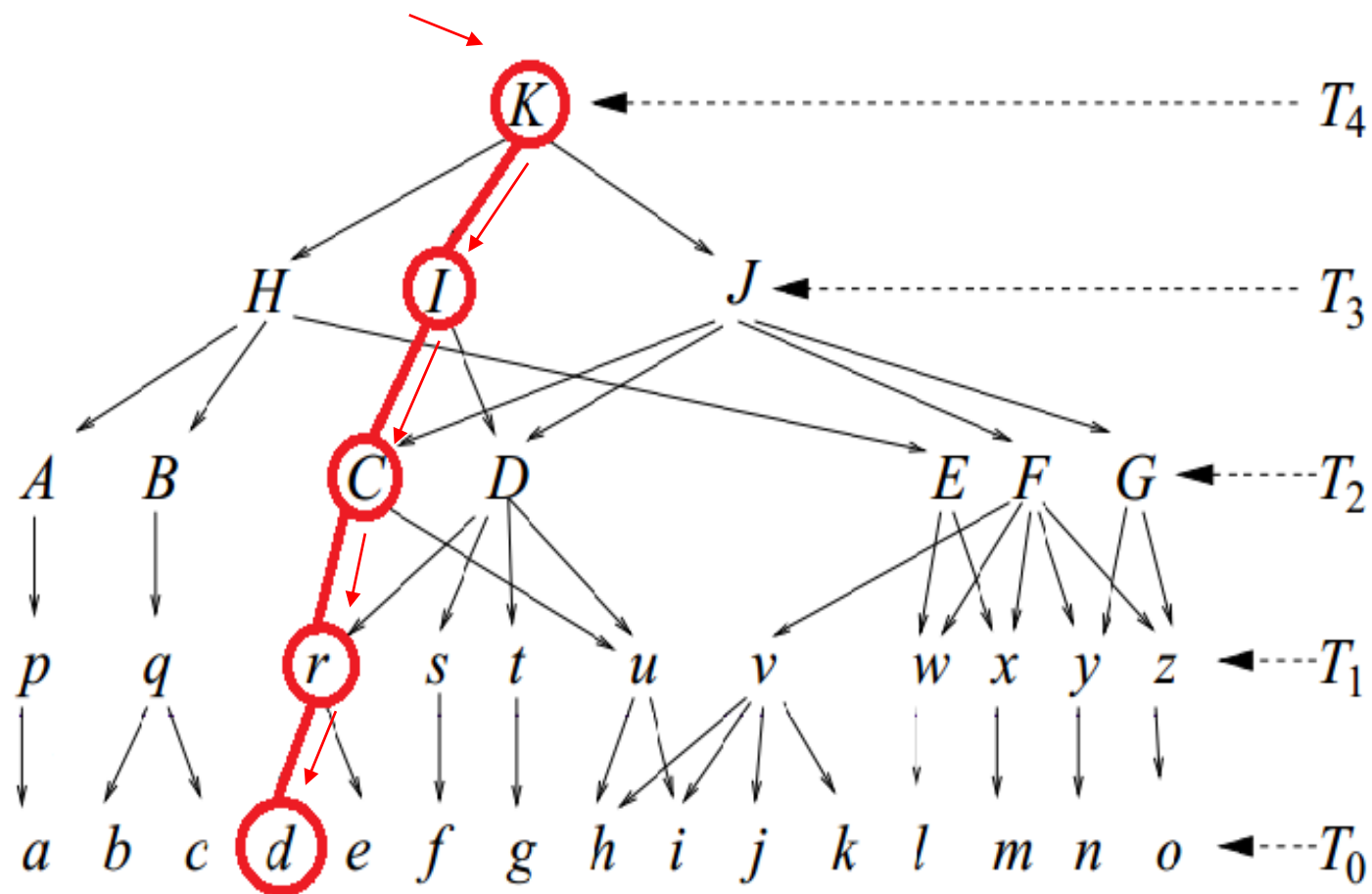
<http://www.cs.ucsb.edu/~suri/cs235/Location.pdf>

Vyhledávání



Obrázek převzat a upraven z:
<http://www.cs.ucsb.edu/~suri/cs235/Location.pdf>

Vyhledávání v grafu



Obrázek převzat a upraven z:

<http://www.cs.ucsb.edu/~suri/cs235/Location.pdf>

Shrnutí

- Sekvence triangulací
- DAG
- Časová náročnost vyhledávání $O(\log n)$
- Paměťová náročnost $O(n)$ – pouze uložení všech vzniklých uzlů a ukazatelů

Zdroje

- <http://www.cs.ucsb.edu/~suri/cs235/Location.pdf>
- http://en.wikipedia.org/wiki/Point_location
- <http://www.cse.iitd.ernet.in/~ssen/cs852/scribe/pointloc/Kirkpatrick.pdf>
- <http://web.natur.cuni.cz/~bayertom/Adk/adk5.pdf>

**Děkuji za pozornost
Otázky?**



**OI-OPPA. European Social Fund
Prague & EU: We invest in your future.**
