



Aljaž Kosmač

Iskanje mostov v grafu

Fakulteta za Matematiko in
Fiziko
12. 1. 2017

Motivacija

Socialna omrežja

- ❖ Imamo skupino ljudi, kjer vsakdo pozna vsaj nekoga
- ❖ Poleg tega lahko neka informacija preide od poljubne osebe do druge po neki poti
- ❖ Ali obstaja takšna povezava med dvema, da bi se ob primeru da se ne bi več poznala (pogovarjala) razbila na dve?
- ❖ Primer uporabe: spletna socialna omrežja(Facebook, Tweeter, ...), politične konference(Združeni narodi, ...)



Motivacija

Letalske povezave

- ❖ Neka letalska družba ima določeno število destinacij
- ❖ Nekdo izračuna katere linije med destinacijami se najbolj izplačajo
- ❖ Predpostavljamo, da lahko iz poljubne destinacije pridemo v drugo po določenih povezavah
- ❖ Ali obstajo takšne povezave med dvema destinacijama, da bi se neka skupina destinacij odrezala od ostalih?

A STAR ALLIANCE MEMBER



Motivacija

Gradnja mostov

- ❖ Za primer vzemimo kakšno otoško državo ali pa “bolj vodnato” mesto.
- ❖ Tako prvi kot drugi morata biti povezana z mostovi, da bi se lahko premikali iz enega dela mesta / otočja do drugega.
- ❖ Recimo, da ima uprava našega kraja nekaj več denarja, ki ga bi želela porabiti za gradnjo mostov. Kje se najbolj izplača postaviti most, da bi ob primeru kakšne nesreče imeli varovalo, da se ne bi zgodilo, da bi nek del otočja postal odrezan od drugih?



Matematični model

- ❖ Prej opisane in podobne probleme običajno predstavimo z grafom.
- ❖ Graf je množica vozlišč in množica povezav.
- ❖ Matematično gledano je most tista povezavo za katero velja, da če jo odstranimo povezan graf razpade na več povezanih komponent oz. graf postane nepovezan.

Algoritem za iskanje mostov

Metoda grobe sile

- ❖ Sprehodimo se čez vse povezave v grafu.
- ❖ Vsakič odstranimo eno povezavo in preverimo ali graf ostane povezan.
- ❖ Razumljiv in intuitiven algoritem, pa vendar prepočasen.
- ❖ Časovna zahtevnost: $O(E * (E + V))$, kjer sta E - velikost množice povezav in V - velikost množice vozlišč.

Algoritem za iskanje mostov

Tarjanov algoritem

- ❖ Robert Tarjan, 1974
- ❖ Časovna zahtevnost: $O(E + V)$, linearna.
- ❖ V osnovi preverja za vsako povezavo ali leži na kakšnem ciklu.

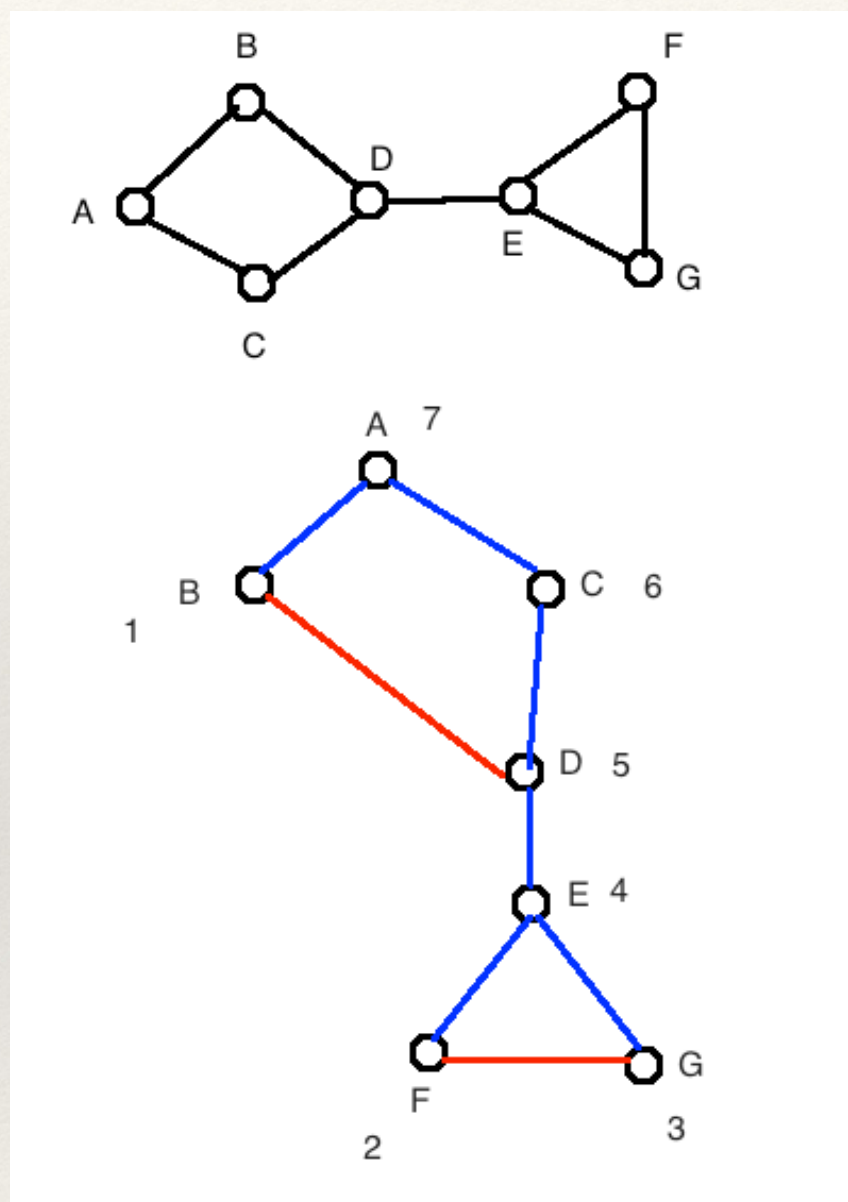
Tarjanov algoritem

Algoritem

- ❖ Za nek graf poišči njegovo drevesno obliko
- ❖ Posebej hrani povezave, ki so del drevesa in tiste, ki niso
- ❖ Z vmesnim pregledom zaporedno oštevilči vozlišča
- ❖ Za vsako vozlišče izračunaj število potomcev
- ❖ Za vsako vozlišče izračunaj najmanjšo število v vmesnem pregledu, ki ga lahko doseže bodisi po drevesni poti s pomikanjem navzdol, bodisi po poti, ki ni del grafa ampak izhaja iz nekega vozlišča, ki je na poti iz danega vozlišča
- ❖ Analogno izračunaj tudi največjo vrednost vmesnega pregleda, ki jo lahko iz nekega vozlišča dosežemo.

Algoritem

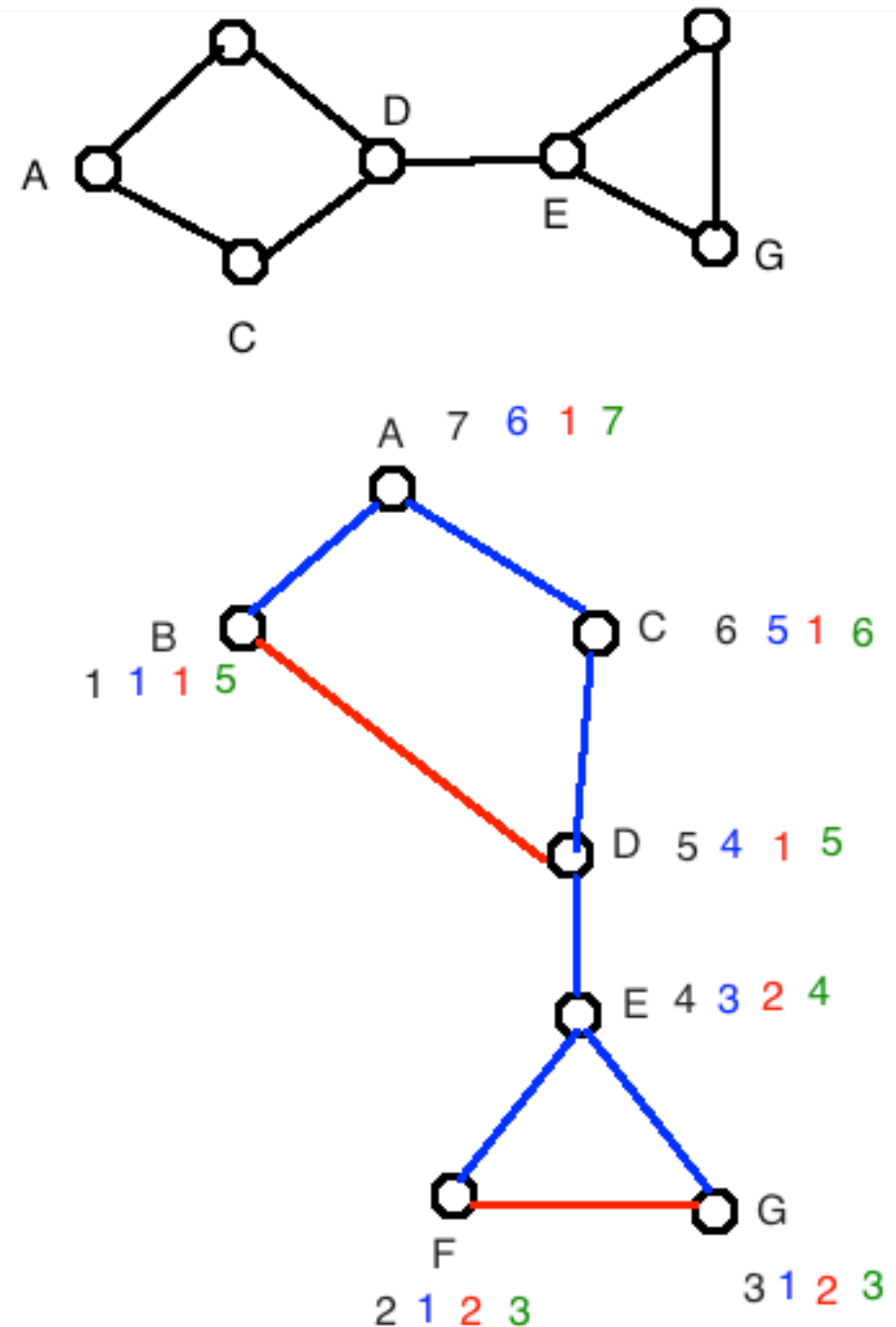
Začetek primera



Tarjanov algoritem

Algoritem

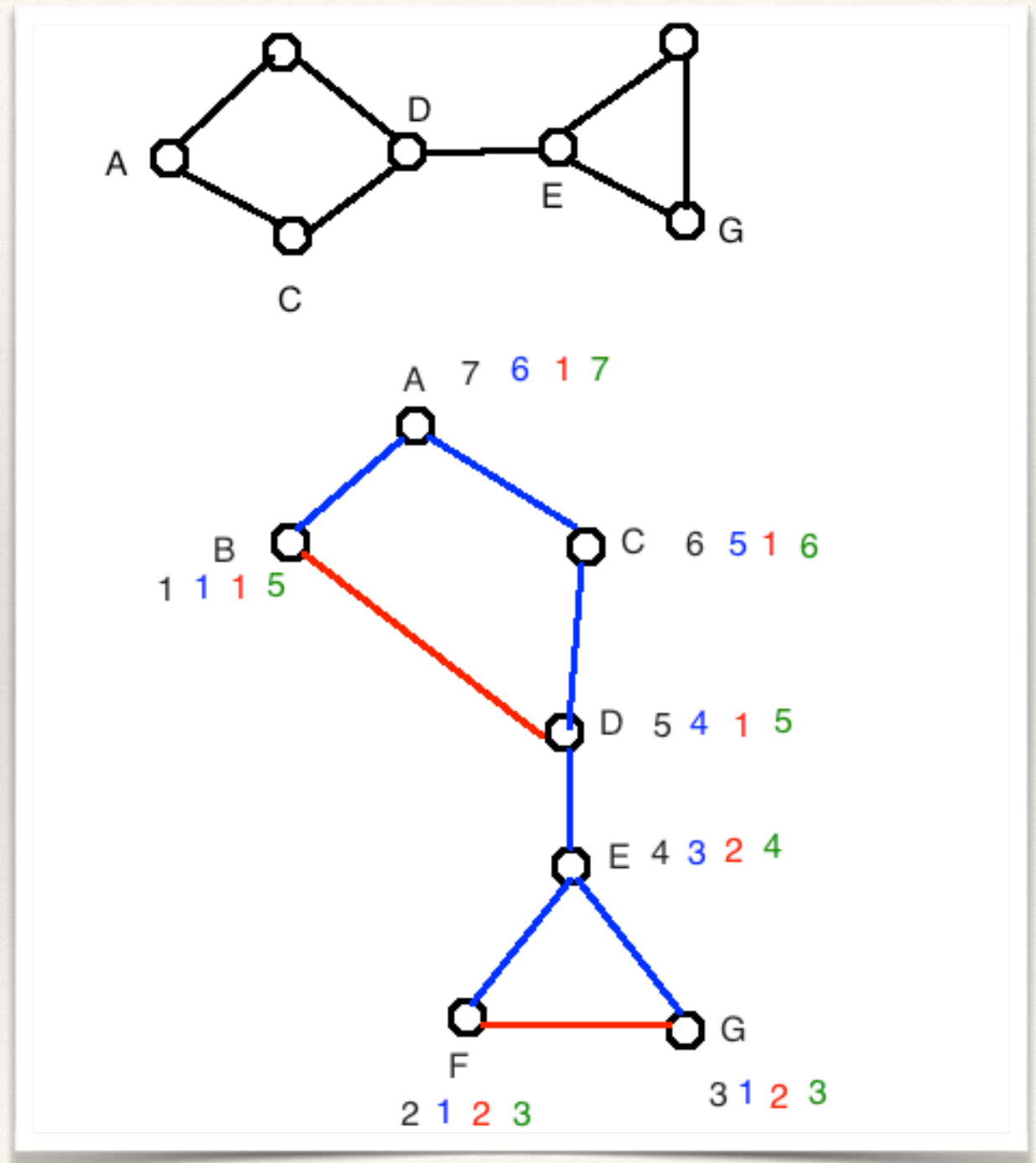
- ❖ Za neko vozlišče v označimo njegovo vrednost v vmesnem pregledu z $\text{pos}(v)$ (črna štečila)
- ❖ Število potomcev nekega vozlišča v označimo z $\text{ND}(v)$ (modra štečila)
- ❖ Najmanjšo vrednost označimo z $L(v)$ (rdeča štečila)
- ❖ Največjo vrednost označimo z $H(v)$ (zelena štečila)



Tarjanov algoritem

Algoritem

- ❖ Povezava, ki izhaja iz očeta vozlišča v , v vozlišče v je most če sta za njo izpolnjena naslednja pogoja:
- ❖ $\mathcal{H}(v) \leq pos(v)$
- ❖ $\mathcal{L}(v) > pos(v) - \mathcal{ND}(v)$
- ❖ Iz primera na desni vidimo, da vozlišče E ustreza našima dvema pogojema.



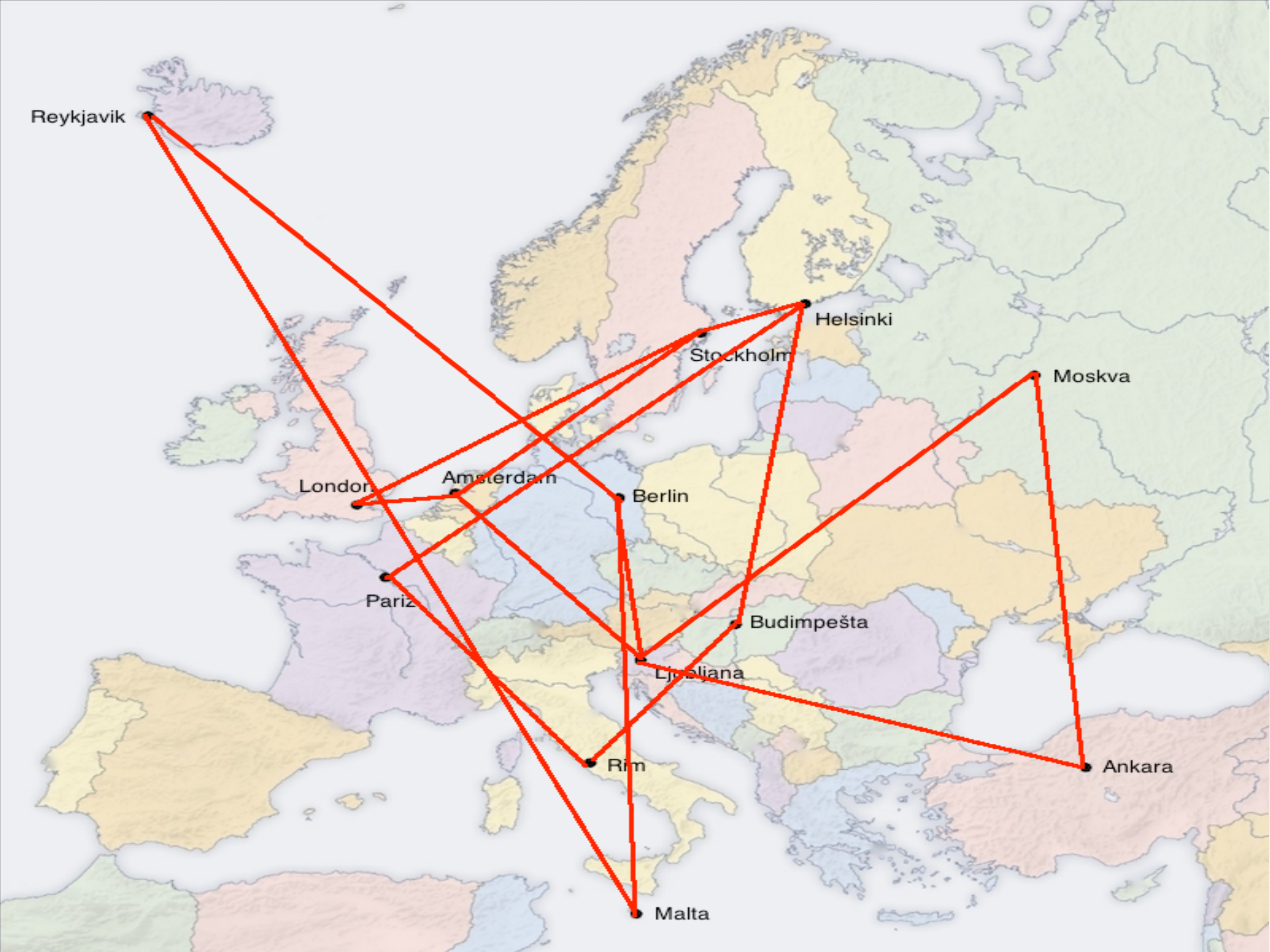
Tarjanov algoritem

Zakaj deluje?

- ❖ Povezave, ki niso del drevesa ne morejo biti mostovi. Očitno saj smo povezavo že dosegli na drevesu, torej se s tem na nek način ponovi.
- ❖ Za vsako vozlišče mora veljati, da je razlika med njegovim številom vmesnega pregleda in njegovim številom potomcev za eno manjša od najnižjega vozlišča v tem poddrevesu.
- ❖ To pa pomeni, da mora biti njegovo število L , tudi v mejah med to omenjeno razliko in pa številom v vmesnem pregledu najnižjega vozlišča v njegovem poddrevesu.
- ❖ Podobno mora veljati tudi, da je najvišja vrednost, ki jo lahko dosežemo manjša ali enaka vrednost vmesnega pregleda v vozlišču.
- ❖ Edina možnost, da to ne bi držalo je ta, da nekje v poddrevesu obstaja povezava, ki ni del grafa in nas popelje ven teh meja, ter se tem ustvari cikel.

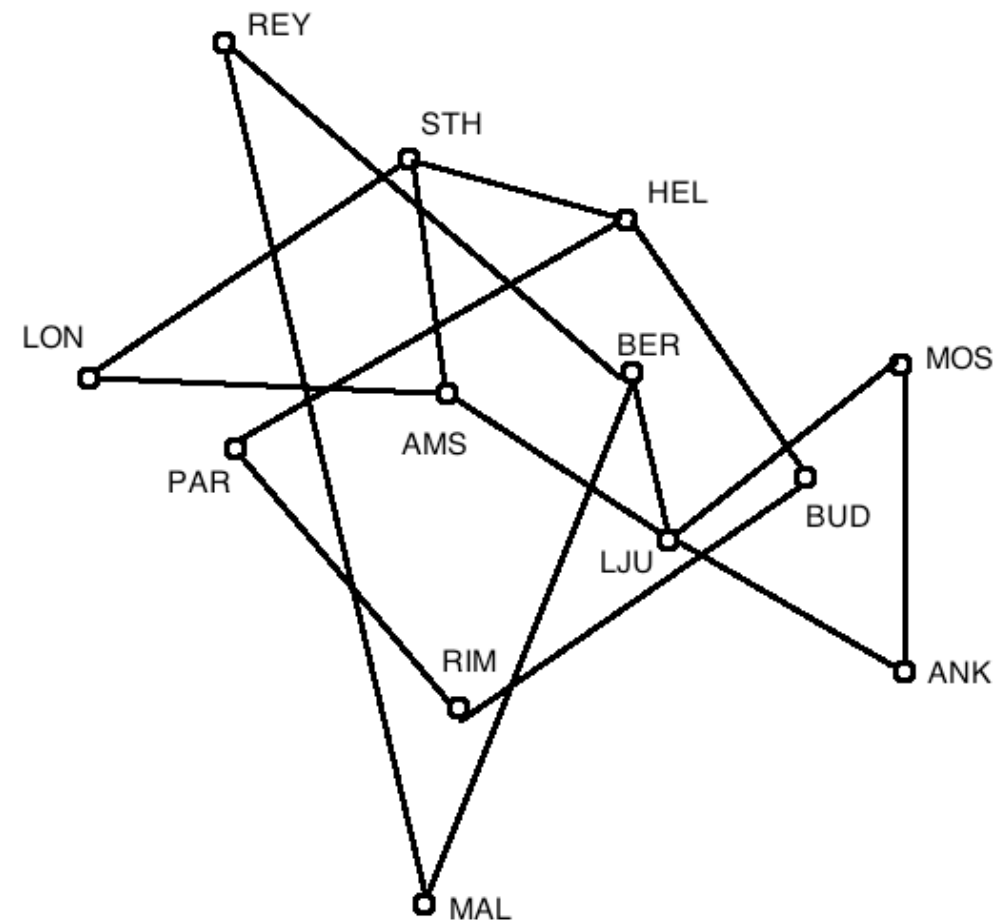
Primer uporabe

- ❖ Ustanovili smo novo letalsko družbo Maϕja.
- ❖ Letalska družba trenutno povezuje 13 Evropskih destinacij.
- ❖ Iz vsake destinacije se da priti na poljubno drugo z največ nekaj menjavami letov.
- ❖ Ali obstaja kakšna povezavo, za katero bi veljalo, da z njeno odstranitvijo, del destenacij postane odrezanih od drugih?



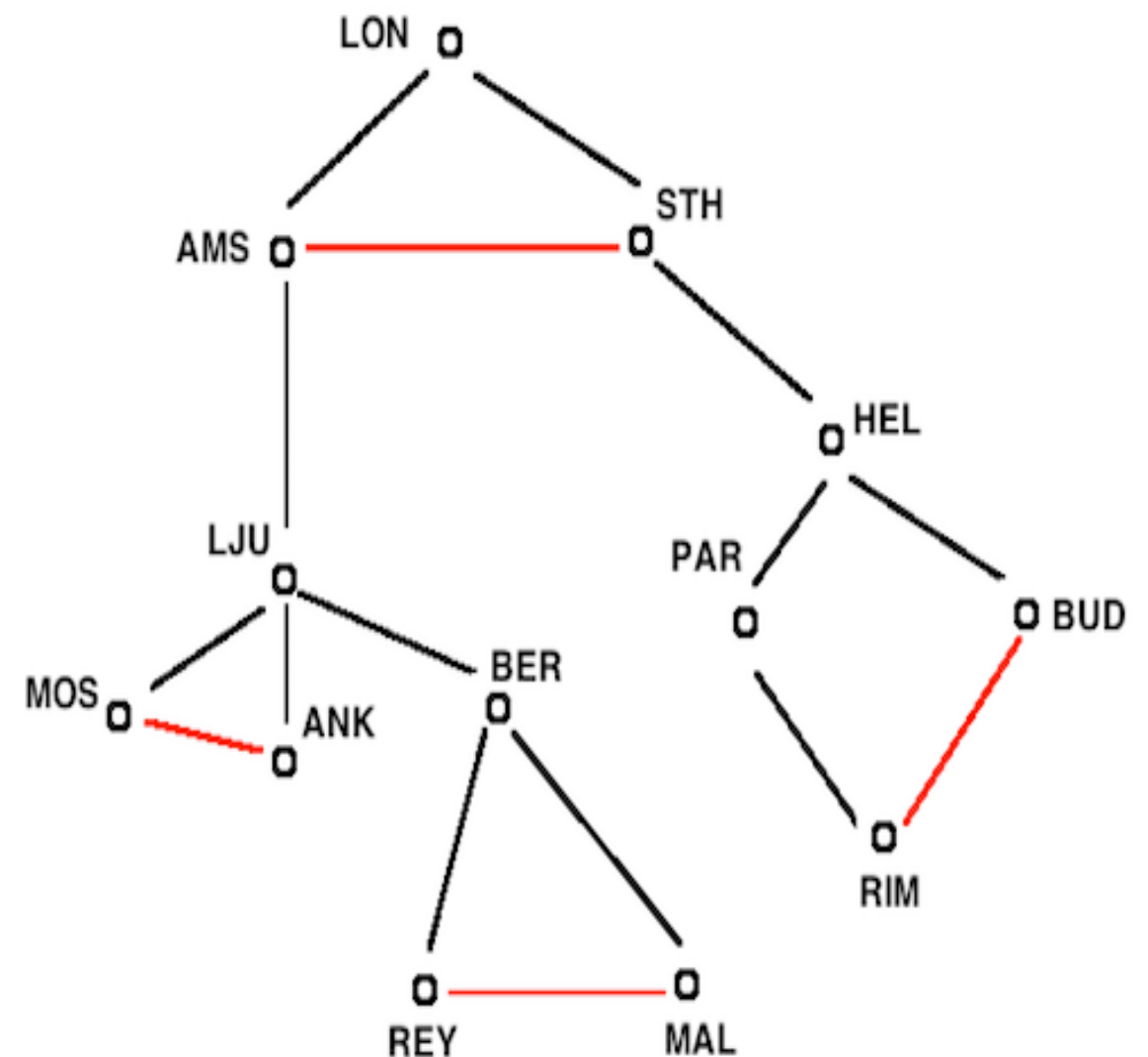
Primer uporabe Matematični model

- ❖ Problem prevedemo na problem iskanja mostov v grafu
- ❖ Iz zemljevida prerišimo povezave in za oznake vozlišč uporabimo njihove kratice
- ❖ Graf na desni je povezan graf na 13 vozliščih



Primer uporabe Drevesna rešitev

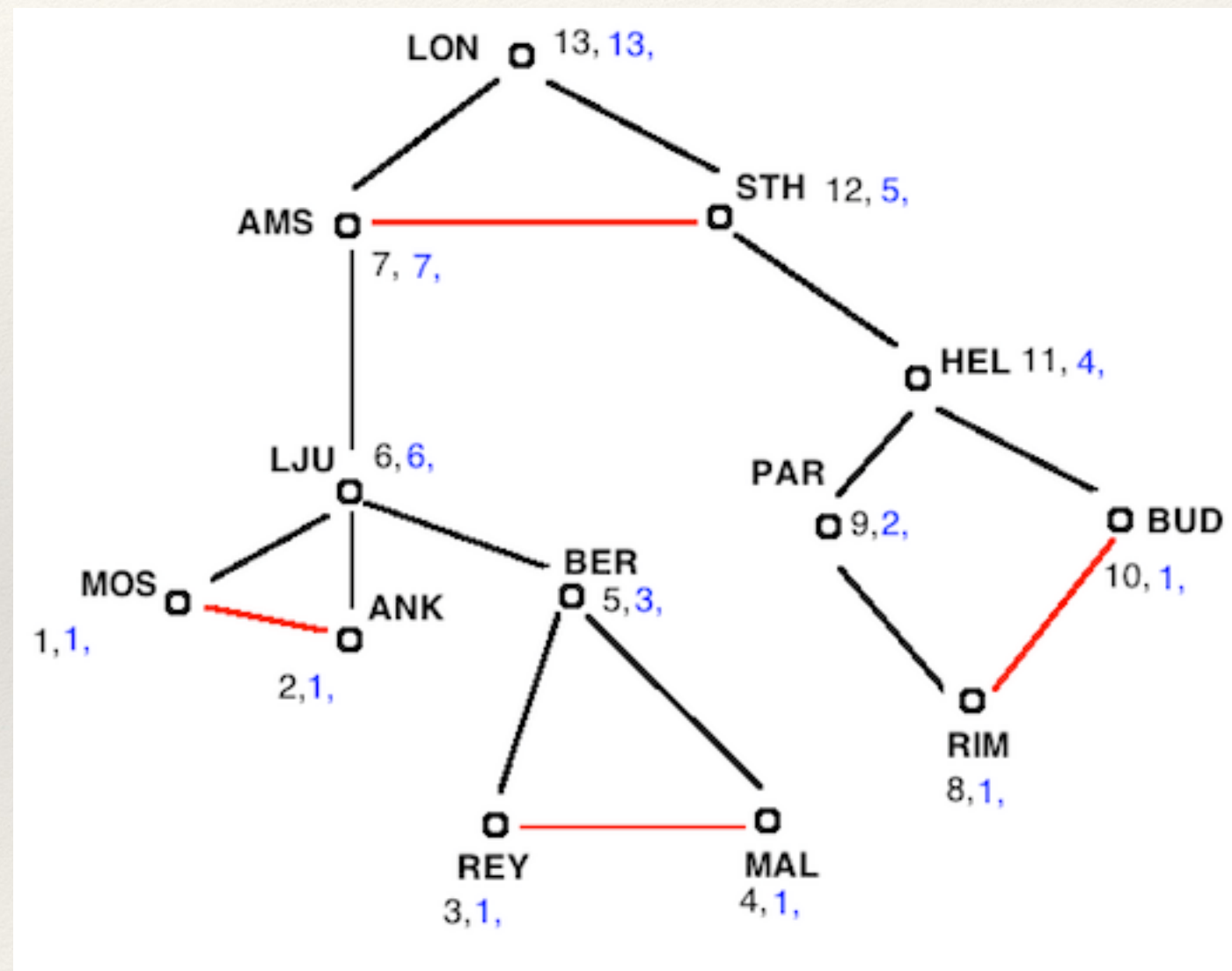
- ❖ Nad grafom izvedemo pregled v širino, da bi dobili drevesno obliko grafa
- ❖ Recimo, da začnemo pri vozlišču z oznako **LON**
- ❖ Črne povezave so del drevesa
- ❖ Rdeče povezave niso del drevesa, so pa del prvotnega grafa



Primer uporabe

Oštevilčenje vozlišč in število potomcev

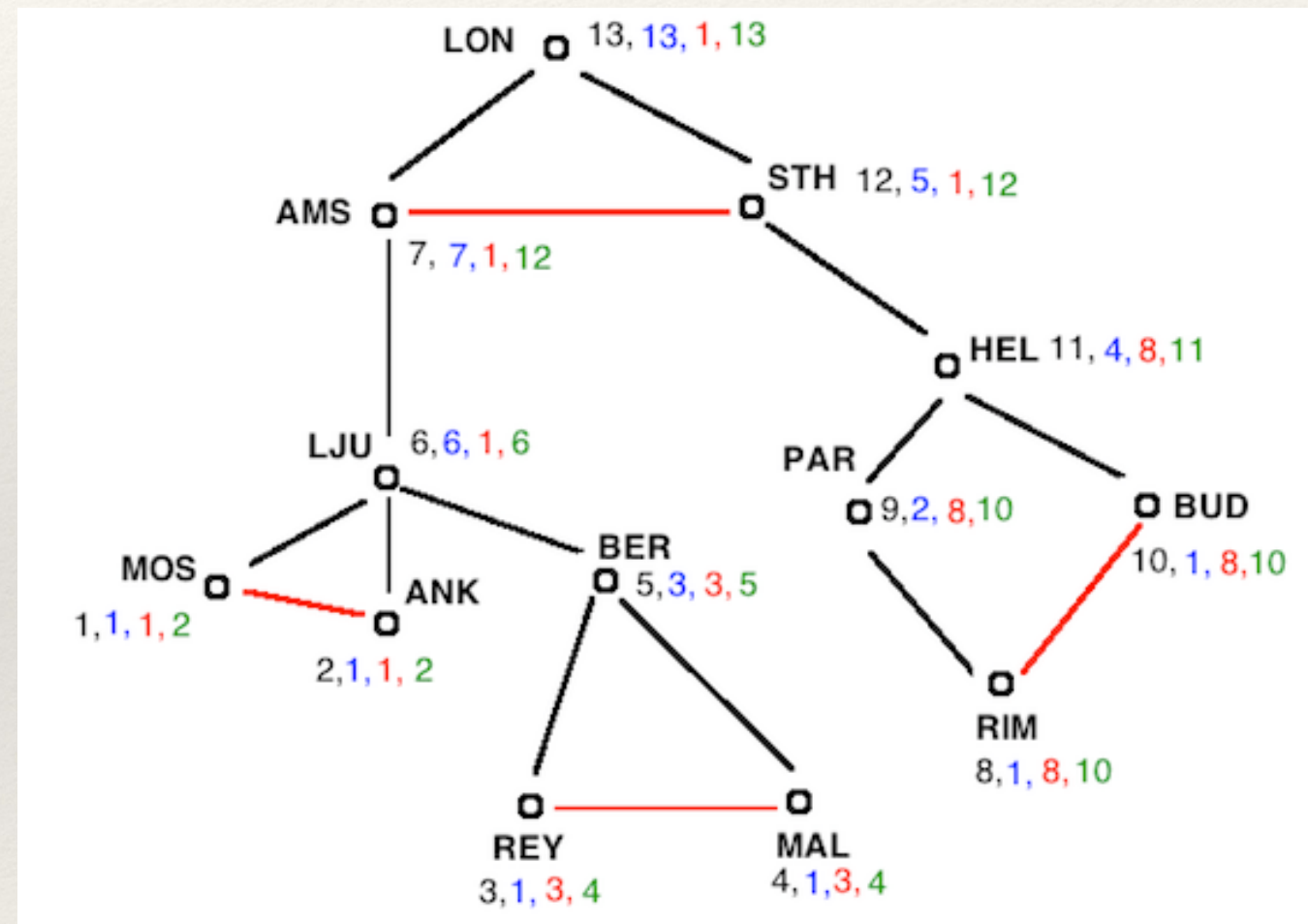
- ❖ Vsakemu vozlišču na grafu določimo zaporedno številko ob vmesnemu pregledu (Črna števila)
- ❖ Vsakemu vozlišču izračunamo število potomcev (Modra števila)



Primer uporabe

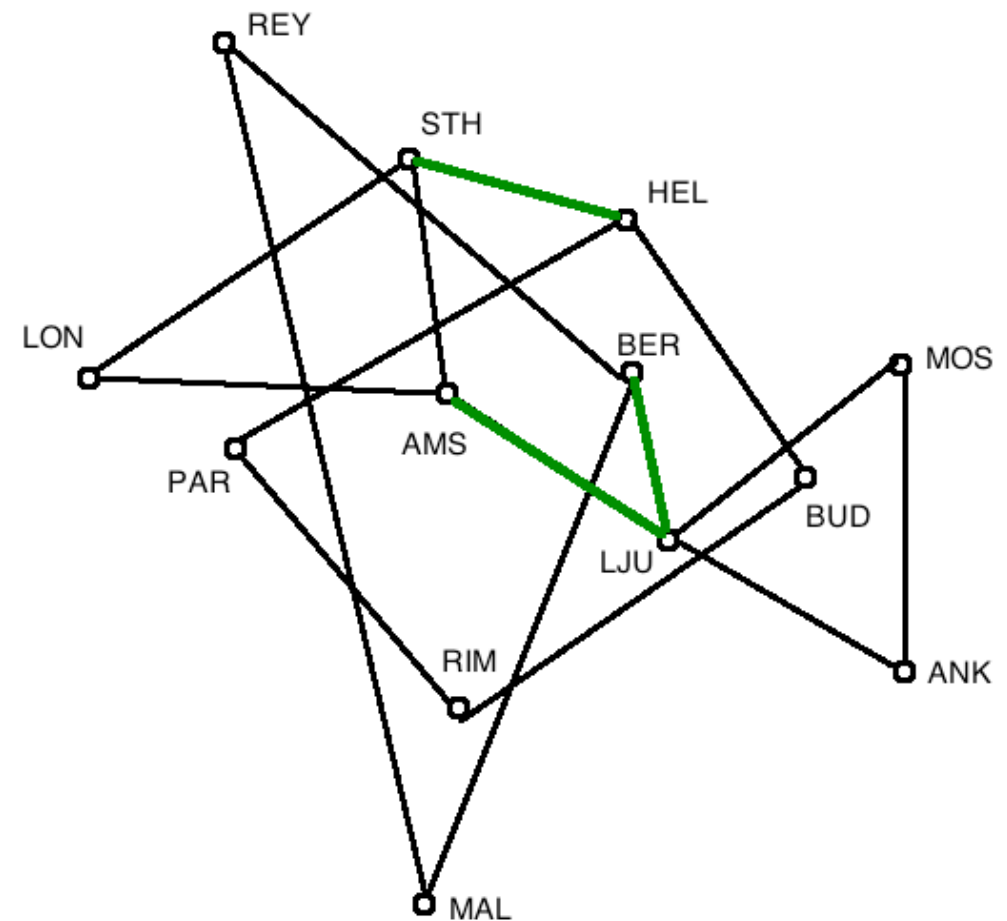
Najmanjše in največje vrednosti

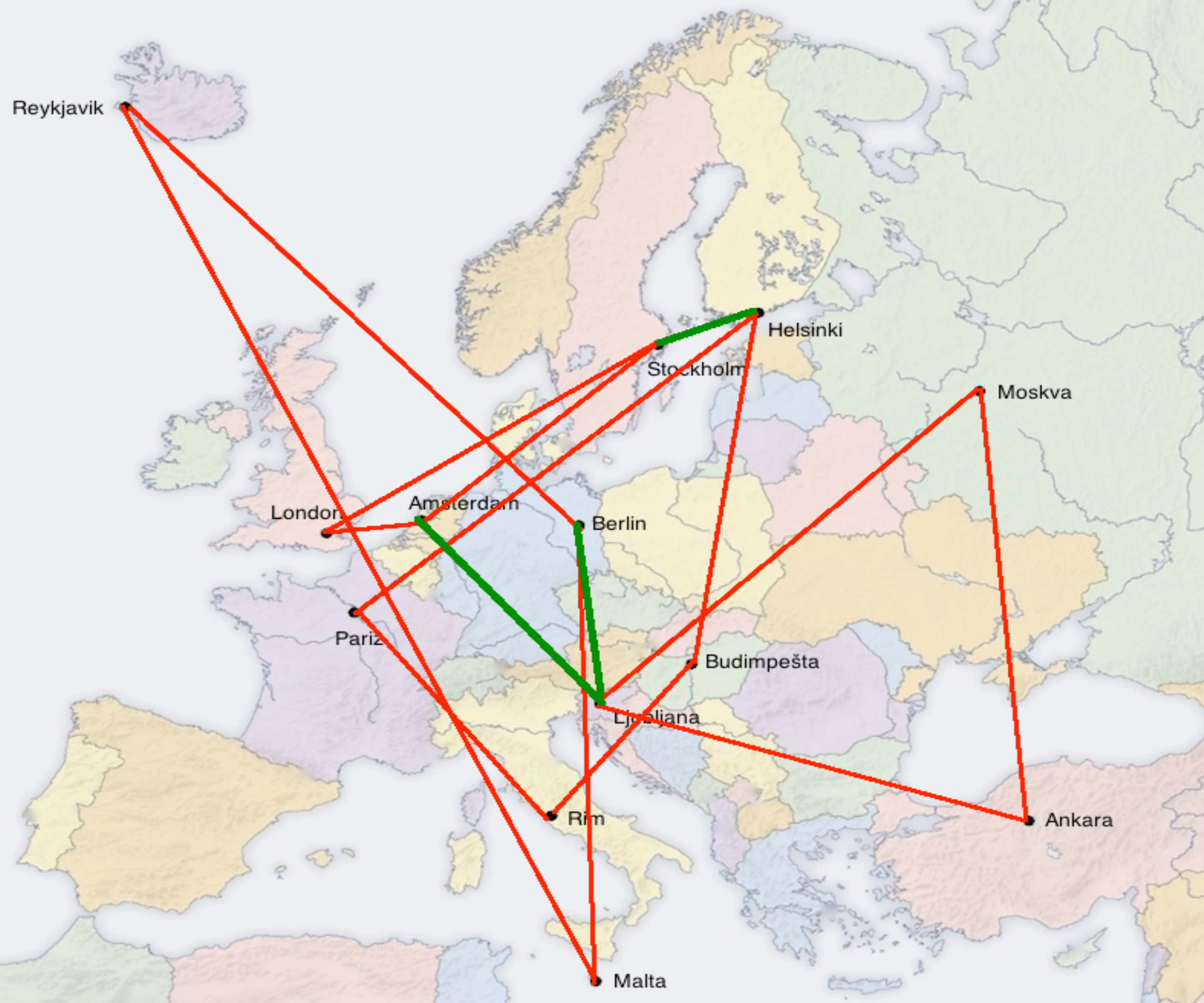
- ❖ Vsakemu vozlišču določimo najmanjšo vrednost vmesenega pregleda, ki jo lahko doseže (Rdeča števila)
- ❖ Analogno določimo tudi največjo vrednost (Zelena števila)
- ❖ Preverimo ali katero vozlišče v ustreza pogoju:
- ❖ $\mathcal{H}(v) \leq pos(v)$
- ❖ $\mathcal{L}(v) > pos(v) - \mathcal{ND}(v)$



Primer uporabe Rešitev

- ❖ Izkaže se, da pogojemata usrezajo vozlišča z oznako **LJU**, **HEL** in **BER**
- ❖ To pomeni, da so mostovi povezave AMS - LJU, LJU - BER in pa STH - HEL
- ❖ V našem kontekstu bi mreža destinacij postala nepovezana če bi ukinili letalske linije med Amsterdamom in Ljubljano, Ljubljano in Berlinom ter Stockholmom in Helsinkimi





O implementaciji

- ❖ Če želimo naš algoritem sprogramirati v npr. Python, bi najprej potrebovali razred, ki bi predstavil graf kot podatkovno strukturo
- ❖ Razmisliti je potrebno, kako v čim manj korakih izračunati vse potrebne podatke o vozlišču
- ❖ Dobro idejo implementacije najdemo na povezavi: <http://www.geeksforgeeks.org/bridge-in-a-graph/>
- ❖ Ta algoritem v osnovi preverja iste stvari kot prej opisani algoritem, le na malo drugačen način.