

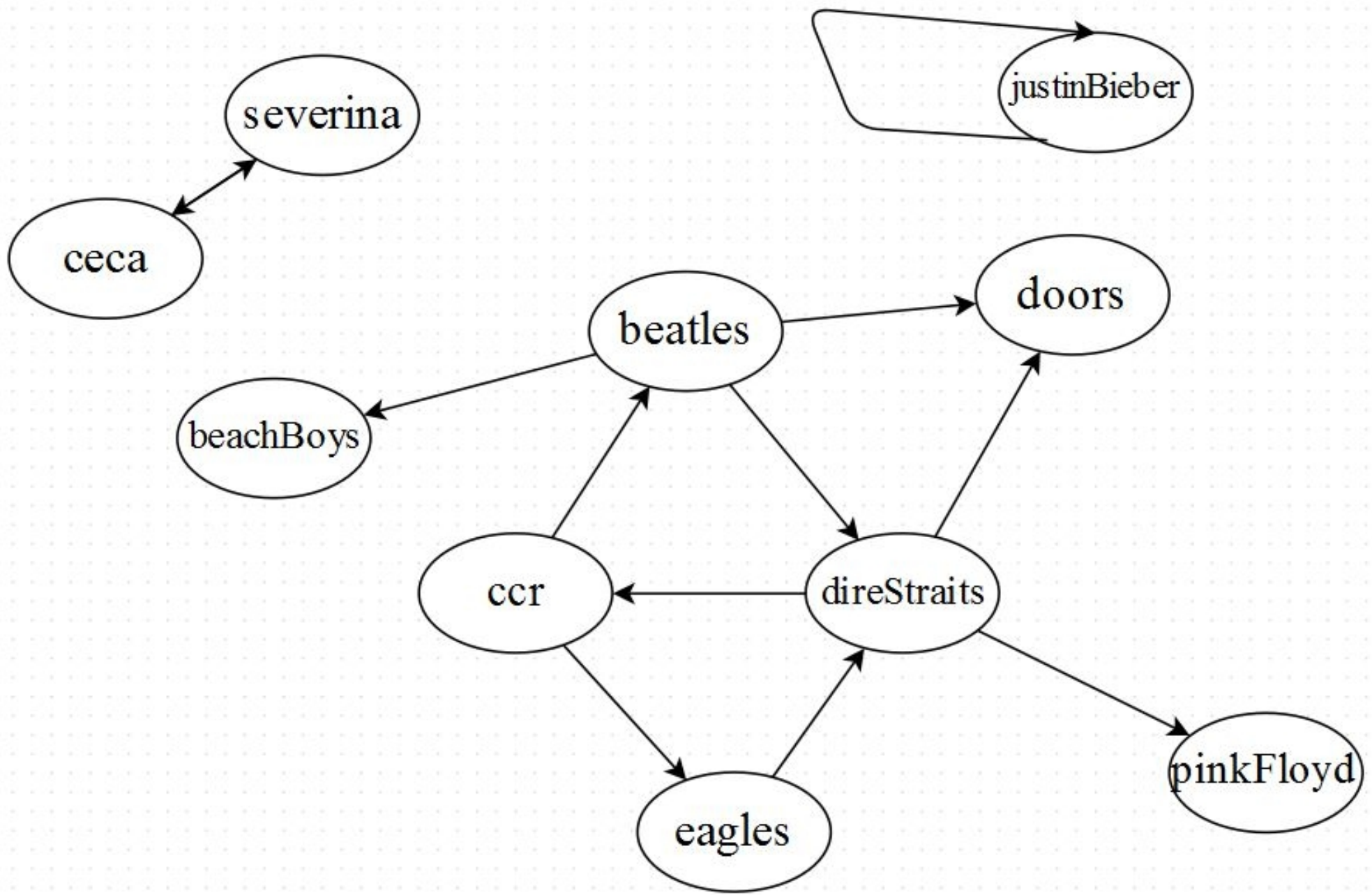
Kosarajujev algoritem

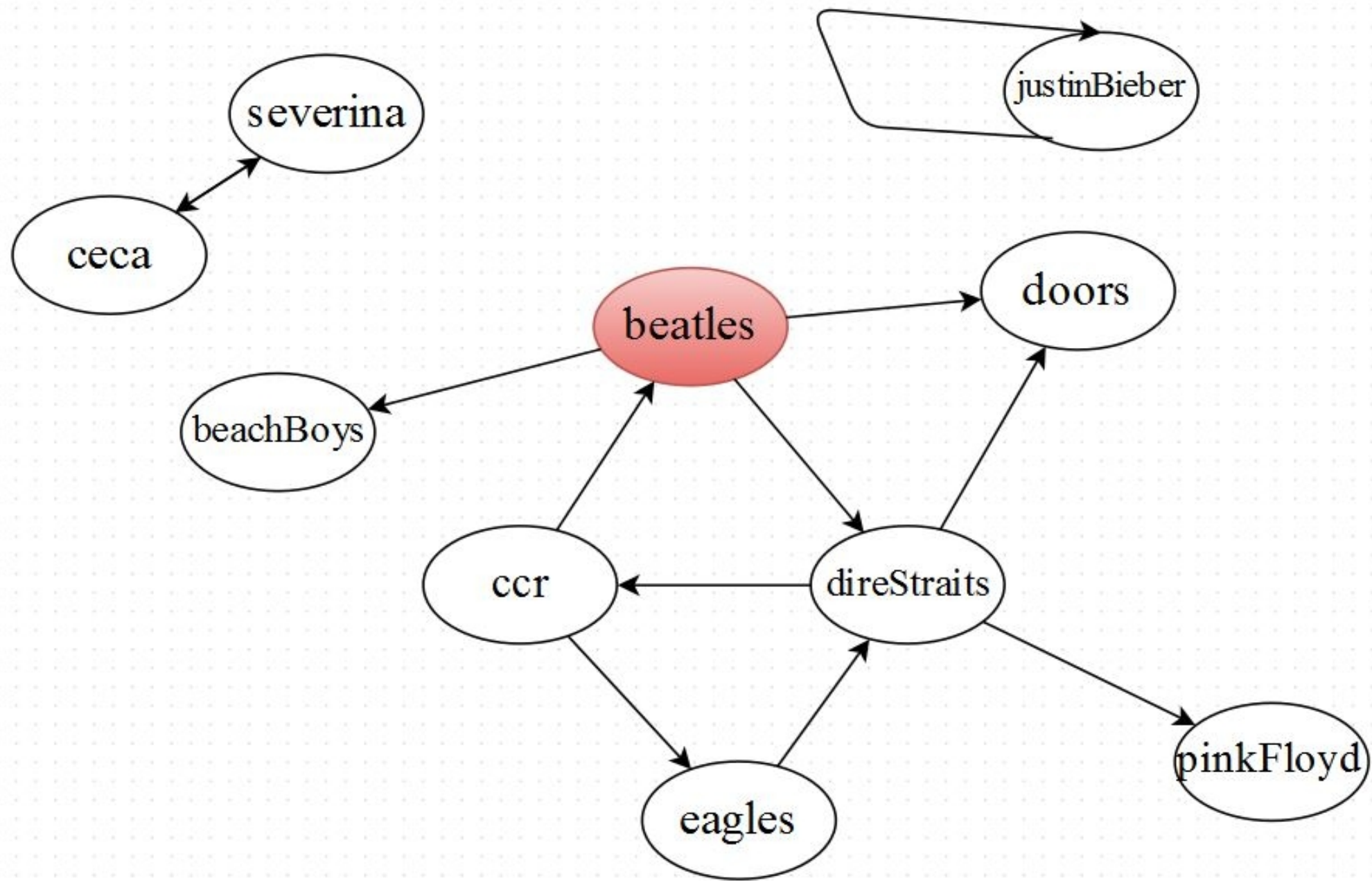
Marko Žugelj
Praktična matematika, 3.letnik
Univerza Ljubljana, FMF
Januar 2016
Mentor: Matija Lokar

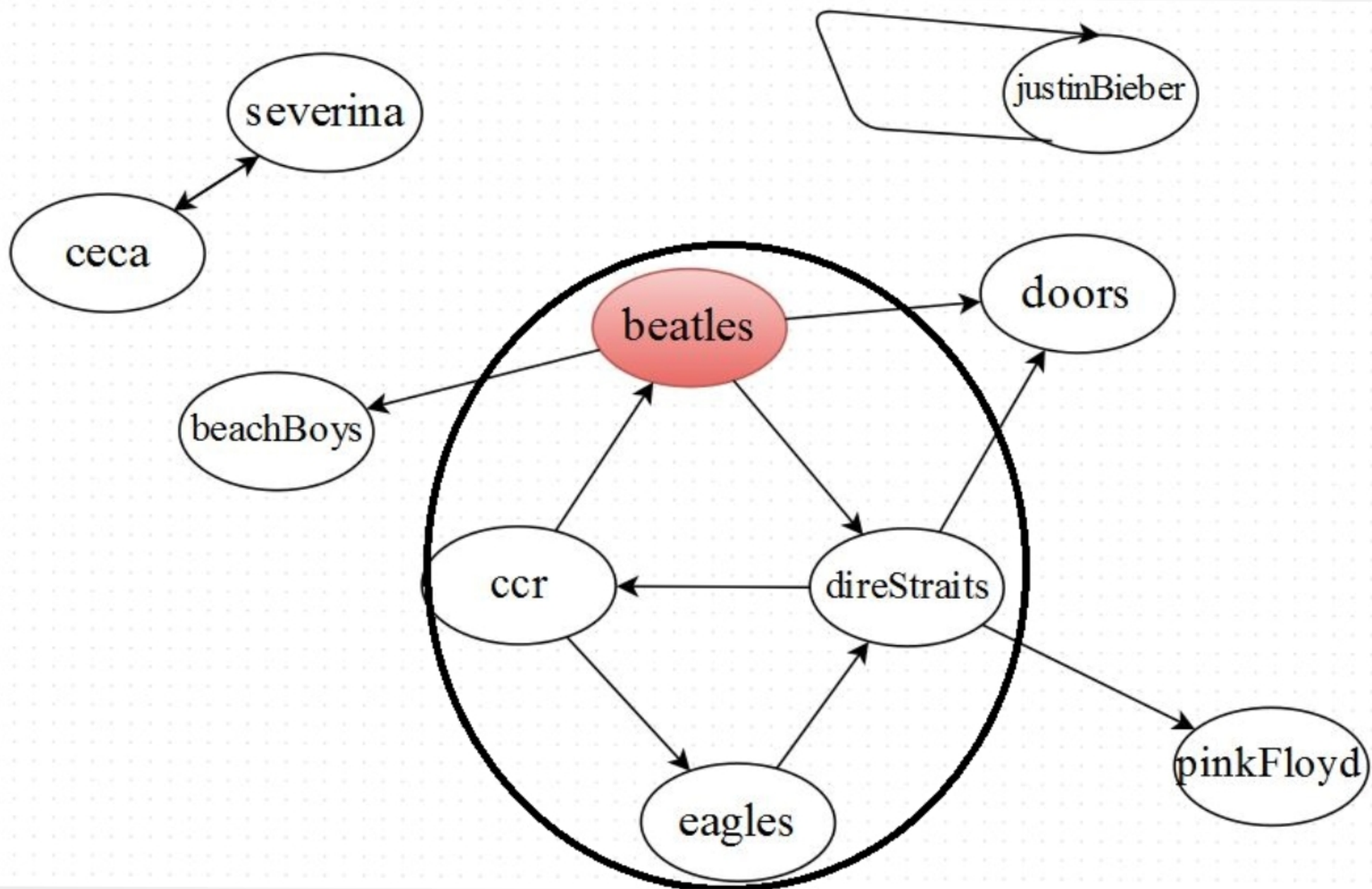
Osnovna motivacija

- Imamo veliko podatkov
- Želimo jih kar najbolje unovčiti
- Priporočeno za vas
- Oglasi
- Morda vas zanima tudi?
- Socialna omrežja:
- Kdo še bi lahko bil naš prijatelj?





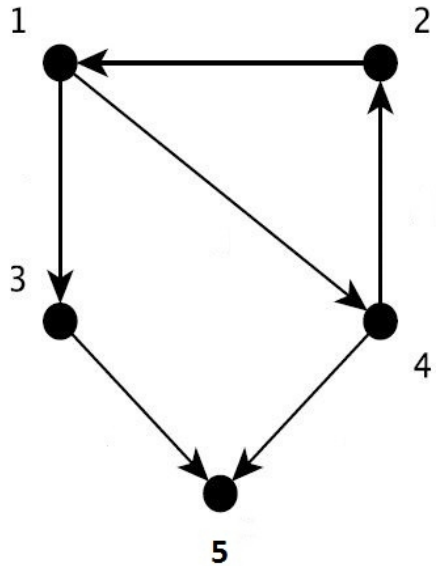




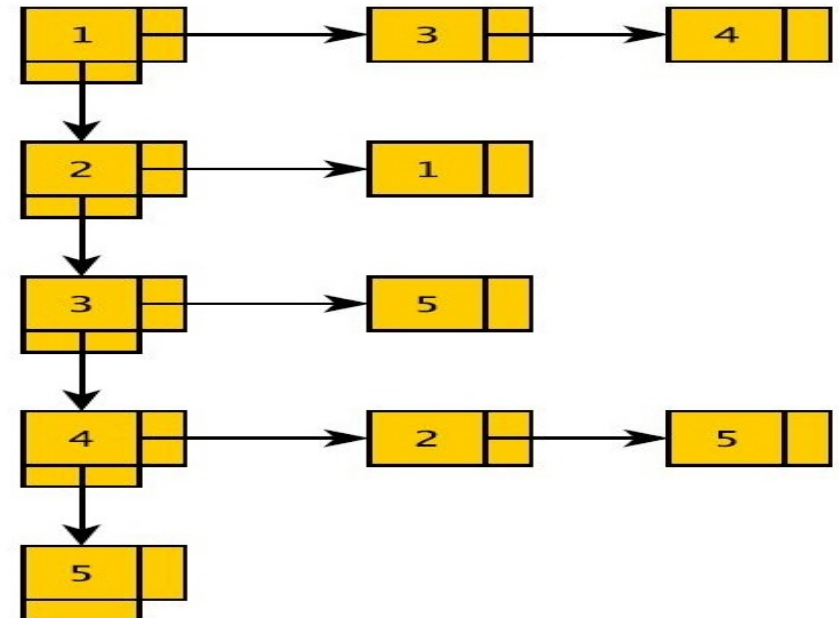
Grafi

- $G=(V,E)$
- Usmerjeni ali neusmerjeni
- Predstavimo z matriko
- Predstavimo s seznamom sosednosti
- Prednosti, slabosti?

Predstavitev s seznamom in matriko



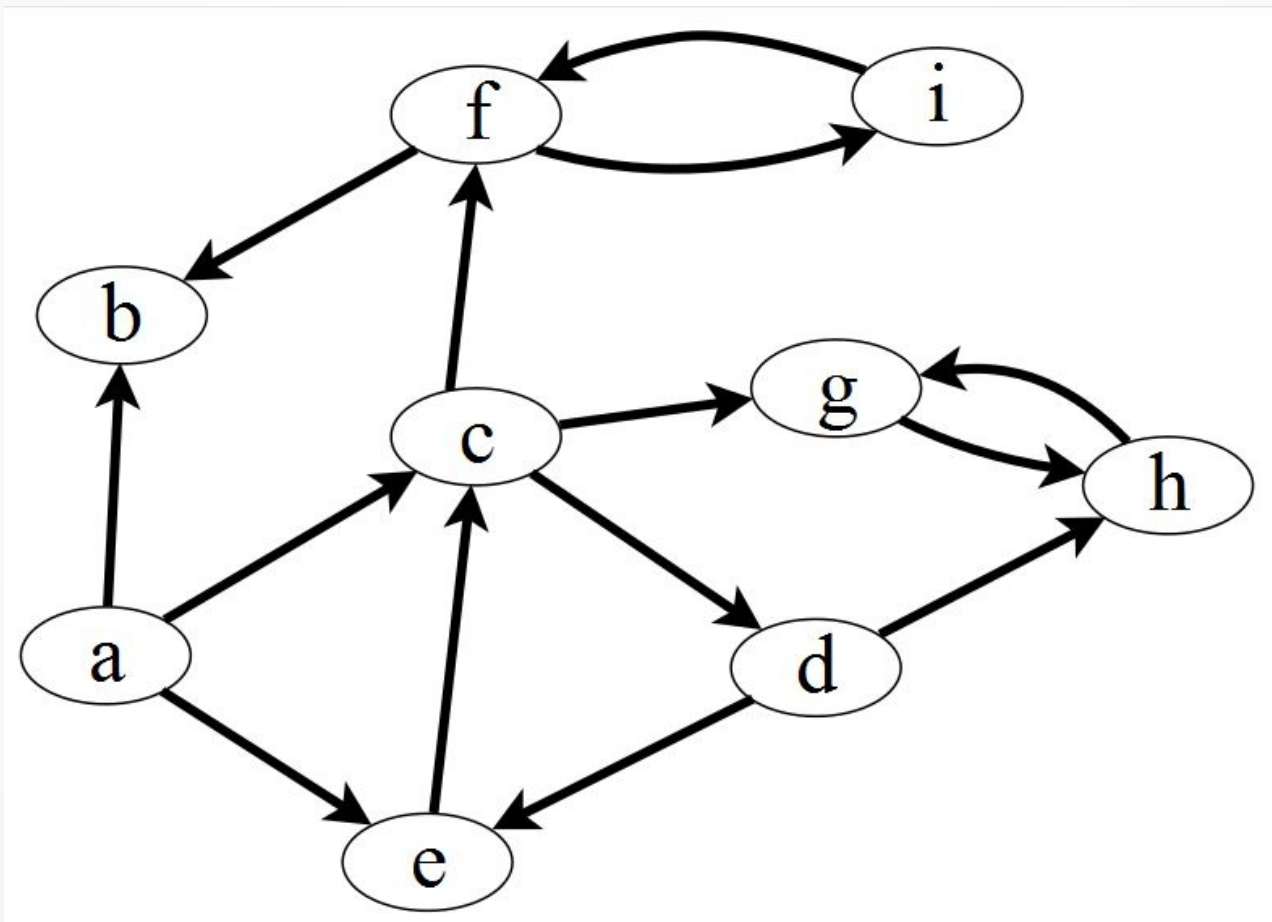
	1	2	3	4	5
1	0	0	1	1	0
2	1	0	0	0	0
3	0	0	0	0	1
4	0	1	0	0	1
5	0	0	0	0	0



Pregled grafa v globino

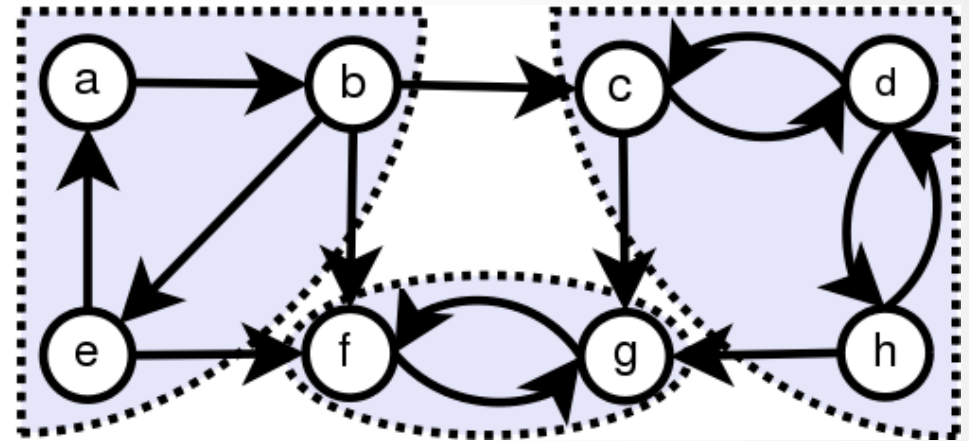
- Izberemo poljubno vozlišče
- Poglejmo njegove sosede
- Če obstaja še neodkrit sosed, izvedi pregled v globino iz soseda
- Za lepšo ponazoritev: vozlišča barvamo
- Belo: neodkrito, zeleno: odkrito, rdeče: pregledali vse njegove sosede
- Včasih beležimo še, kdaj smo odkrili vozlišče, kdaj smo pregledali vse njegove sosede
- Pregled v globino tvori drevo (gozd)
- Ali se da brez rekurzije?

Zgled

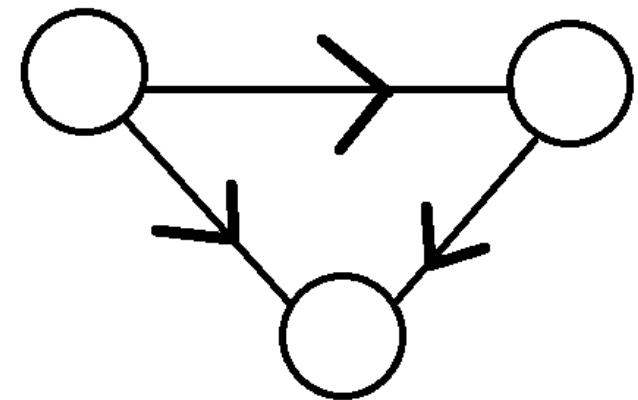


Krepko povezane komponente

- Podmnožica vozlišč
- Za vsaki vozlišči u in v :
 - Obstaja pot od u do v
 - Obstaja pot od v do u
 -

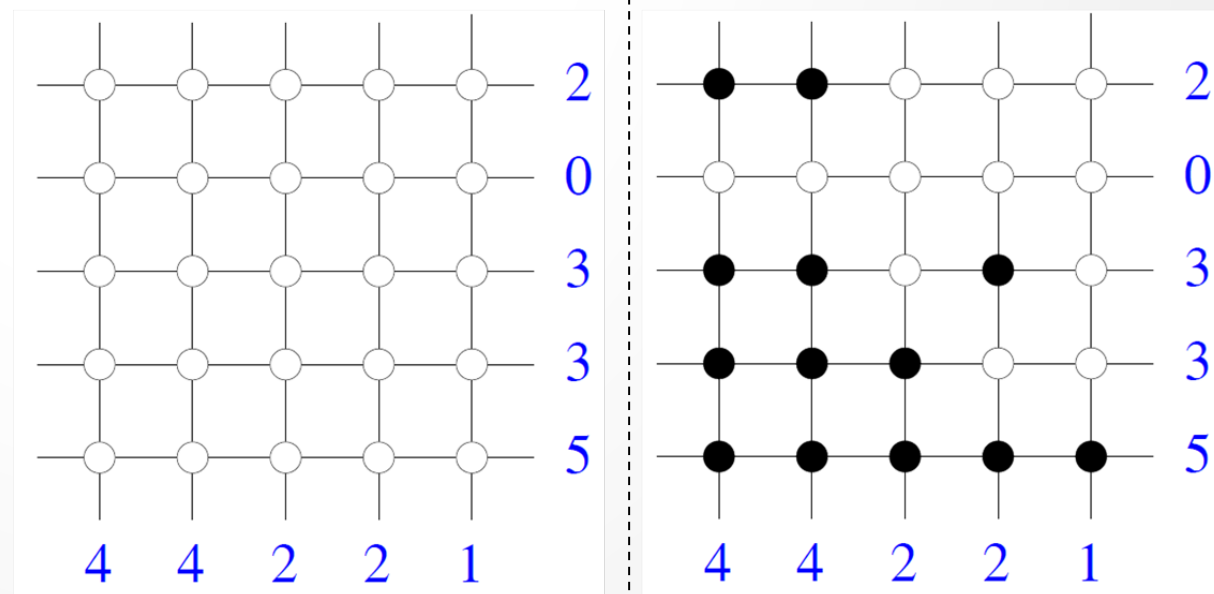


- Obstaja cikel? Kaj pa obratno?
- Kaj če dodamo povezavo?
- Komponentni graf: krepko povezana komponenta $\rightarrow$ vozlišče
- Komponentni graf je brez ciklov



Uporaba krepko povezanih komponent

- Deli in vladaj
- Analiza socialnih omrežij
- Reševanje 2-SAT problemov
- Optimizacija prevajalnikov
- Iskanje »zastojev« v operacijski sistemih in paralelnem programiranju



Optimizacija prevajalnikov

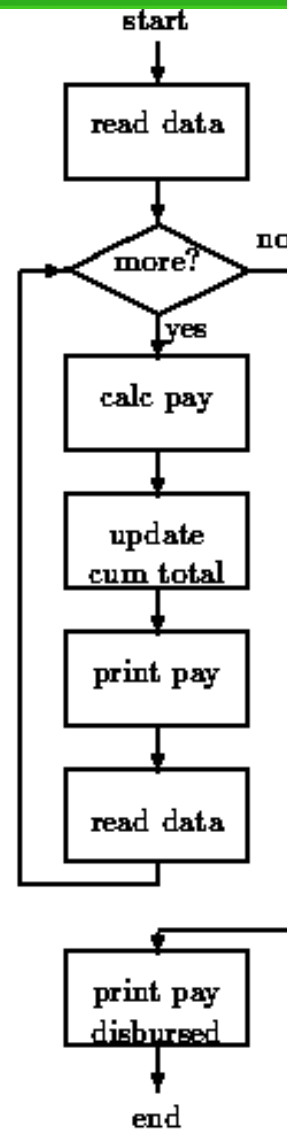
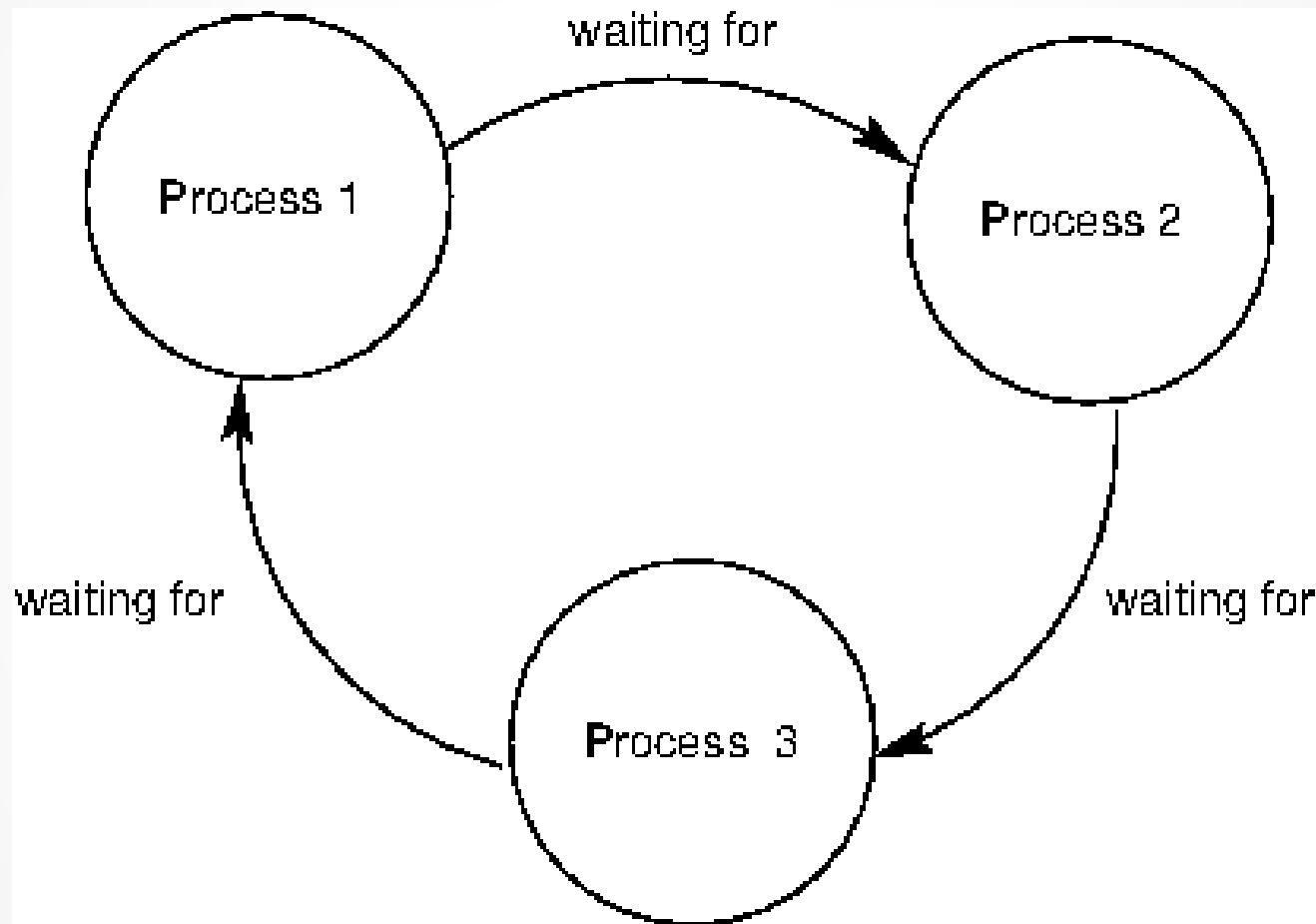
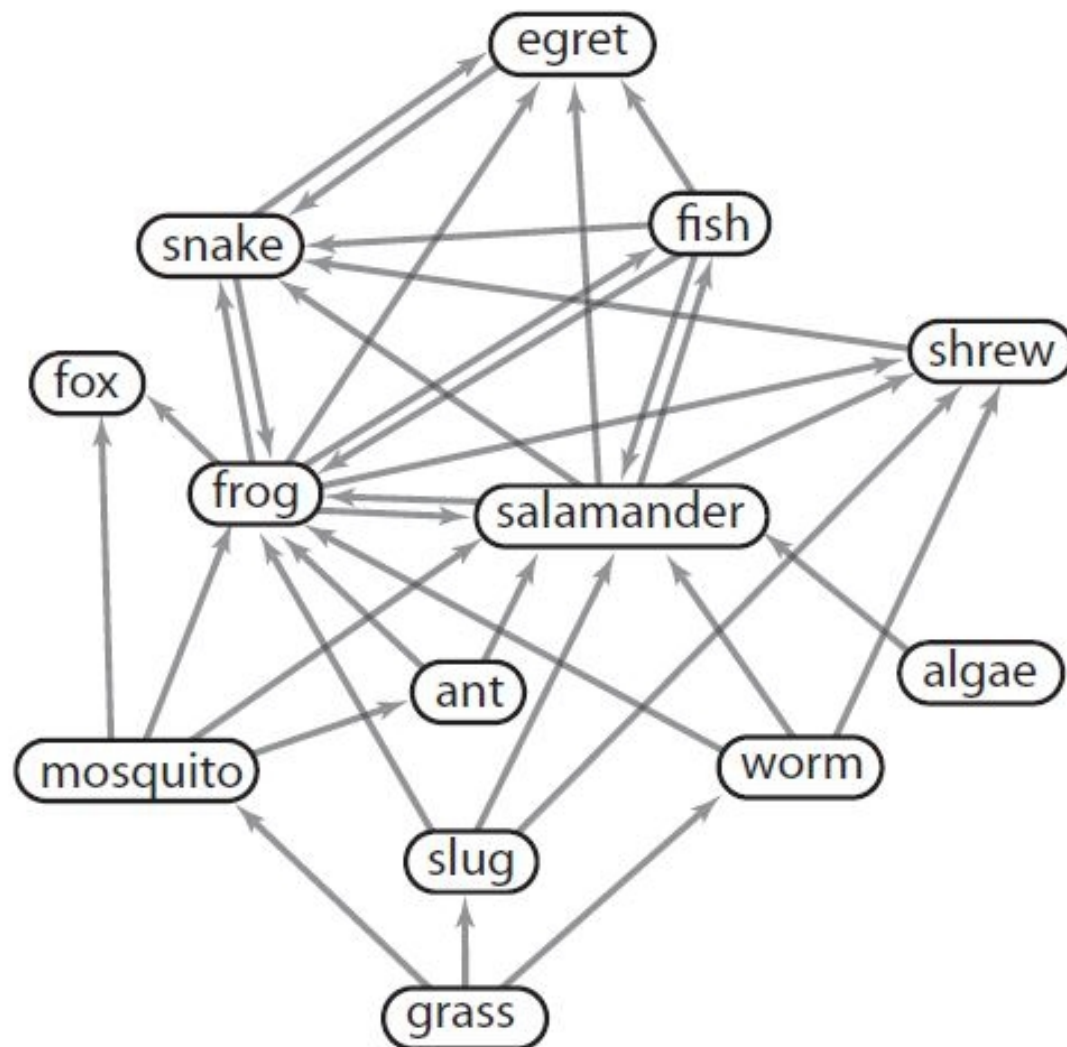


Figure 1.6: Flow Chart for Payroll Task

Zastoji v paralelnem programiranju



Analiza ekosistemov



Transponiran graf

- Obrnemo vse povezave
- Krepko povezane komponente so iste
- Def: Končni čas komponente je največji čas med končnimi časi vozlišč

- **Izrek:** povezave med krepko povezanimi komponentami v **transponiranem** grafu potekajo le od komponente z manjšim končnim časom proti tisti z večjim končnim časom.

Kosarajujev algoritem

- Izvedemo pregled grafa v globino
- Poiščemo transponiran graf
- Izvedemo pregled transponiranega grafa v globino, kjer izbiramo vozlišča po padajočem končnem času.
- Vrnemo vsako drevo iz gozda kot krepko povezano komponento

O algoritmu

- 1978 Kosaraju
- 1981 Sharir
- Linearne časovne zahtevnosti
- Tarjan

Literatura

- http://4.bp.blogspot.com/-YFsZ_bDGqNc/UutrTEZJI ZI/AAAAAAAAA-s/6s9W8hQ7Gag/s1600/deadlock.png
- T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to Algorithms*, 3rd Edition, MIT Press, 2009, poglavje 22.
- https://en.wikipedia.org/wiki/Kosaraju's_algorithm (januar 2016)
- https://en.wikipedia.org/wiki/Strongly_connected_component (januar 2016)
- <https://en.wikipedia.org/wiki/2-satisfiability> (januar 2016)