

# DISTRIBUIRANI ALGORITMI I SISTEMI

# A1 Slanje svima preko fiksiranog razapinjućeg stabla (1 / 2)

2

- koren inicijalno šalje  $M$  svojim potomcima
- kada procesor primi  $M$  od svog predka
  - ▣ šalje  $M$  svojim potomcima
  - ▣ završava rad (postavlja lokalnu bool prom. na true)

Vežba: transformisati ovaj pseudokod u opis u stilu automata

# A1 Slanje svima preko fiksiranog razapinjućeg stabla (2/2)

3

- Neka procesori već imaju info. o nepokretnom razapinjućem stablu date komunikacione topologije
  - ▣ stablo: povezan graf bez petlji
  - ▣ razapinjuće stablo: sadrži sve procesore
  - ▣ fiksirano: postoji jedinstven korenski čvor
- Implementacija putem lokalnih prom. *parent* i *children* na svakom procesoru
  - ▣ ukazuju koji incidentni kanali vode do predka i potomka u fiksiranom razapinjućem stablu

# Analiza složenosti

4

- Sinhron model:
  - vreme jednako dubini razapinjućeg stabla, a ona može biti najviše  $n - 1$
  - broj poruka je  $n - 1$ , pošto se jedna poruka šalje po svakom luku razapinjućeg stabla
- Asinhron model:
  - isto vreme i broj poruka

# Prijem od svih čvorova

5

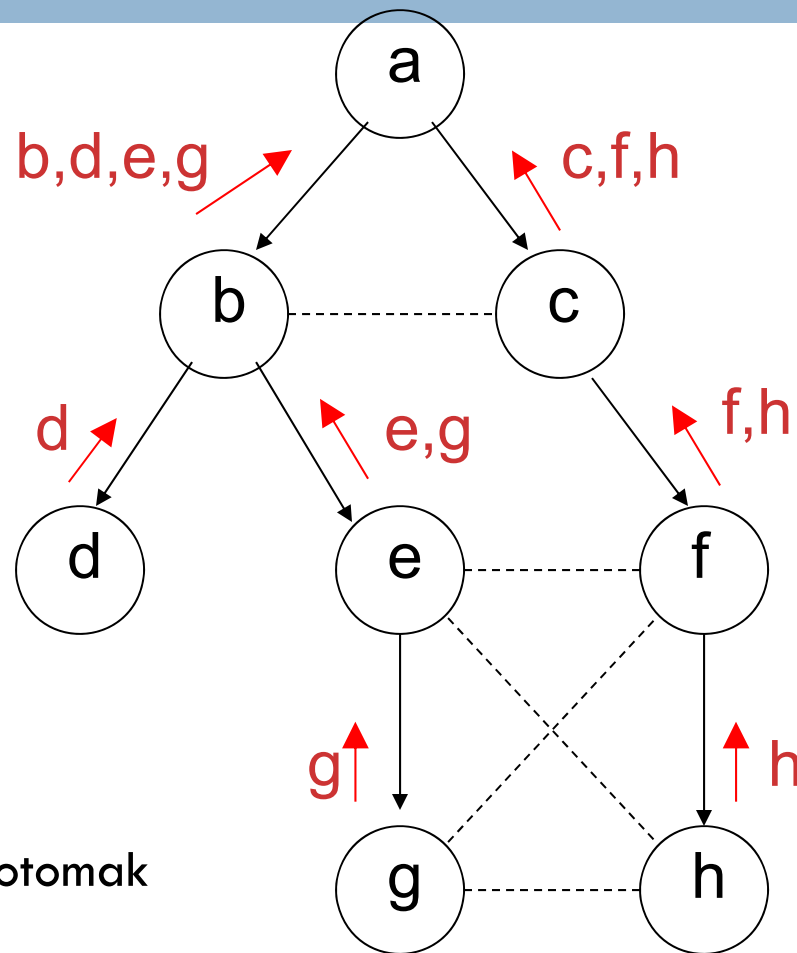
- Predpostavimo da su procesori već odredili fiksirano razapinjuće stablo
  - postavljene su promenljive *parent* i *children* na svakom procesoru
- Radi obrnuto od slanja svima:
  - listovi šalju poruke svojim pretcima
  - čvorovi (koji nisu listovi) čekaju prijem poruke od svih potomaka, zatim šalju kombinovanu info svom predku

# Prijem od svih čvorova

6

isprekidane linije:  
lukovi van stabla

pune strelice:  
relacije predak-potomak



# A2 Pronalaženje razapinjućeg stabla za zadati koren (1 / 2)

7

- Vrlo je pogodno imati razapinjuće stablo.
- Kako ga dobiti?
- Predpostavimo da je poznat neki istaknuti čvor, koji će služiti kao koren stabla.
- Modifikujmo algoritam plavljenja.

# A2 Pronalaženje razapinjućeg stabla za zadati koren (2/2)

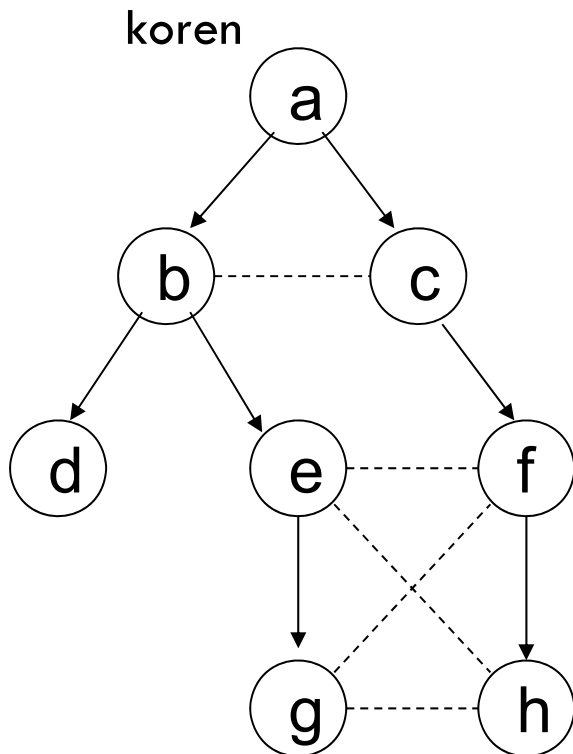
8

- koren šalje  $M$  svim svojim susedima
- kada čvor (koji nije list) *prvi put* primi  $M$ 
  - ▣ postavi pošiljaoca kao svog predka
  - ▣ Pošalji poruku "parent" ka pošiljaocu
  - ▣ pošalji  $M$  ka svim drugim susedima (ako nema drugih suseda, onda završi rad)
- kada primi  $M$  inače (tj. posle prvog puta)
  - ▣ Pošalji poruku „already" ka pošiljaocu
- koristi poruke "parent" i „already" za postavljanje prom *children* i za odluku o završetku rada (posle javljanja svih suseda)



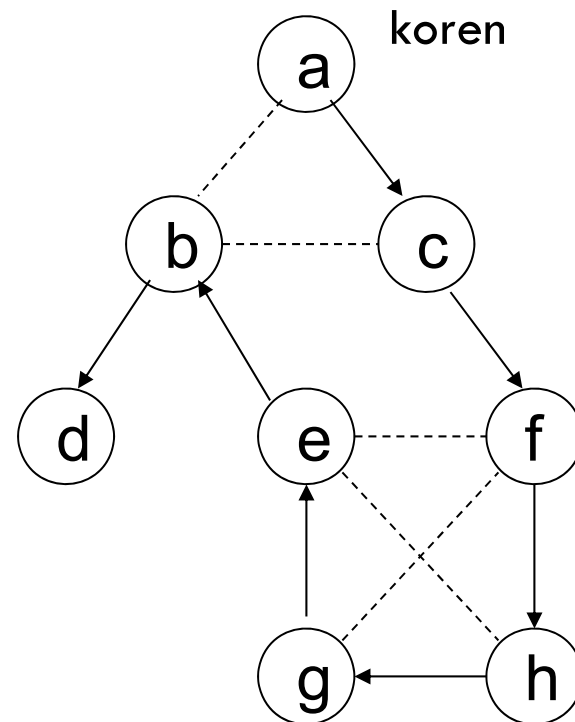
# Izvršenje algoritma razapinjućeg stabla

9



Sinhrono: uvek daje BFS  
(breadth-first search) stablo

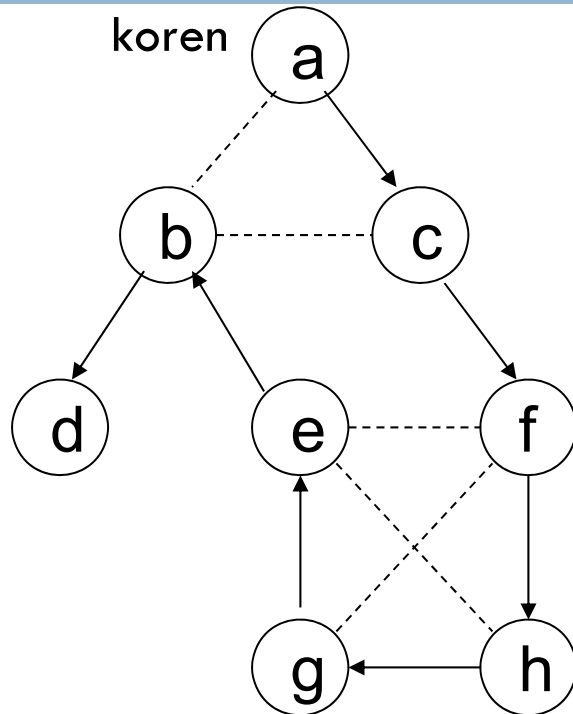
Oba modela:  
 $O(m)$  poruka  
 $O(\text{diam})$  vreme



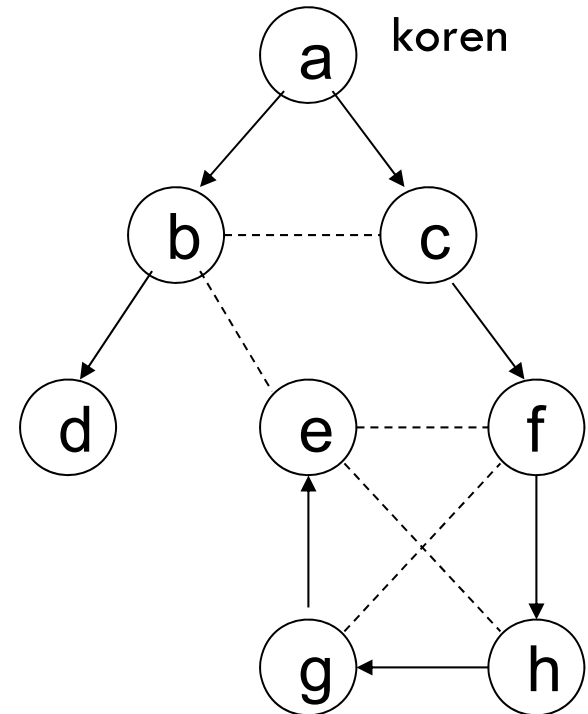
Asinhrono: ne mora  
obavezno dati BFS stablo

# Izvršenje algoritma razapinjućeg stabla

10



Ne!



Asinhron izvršenje je dalo ovo DFS (depth-first search) stablo. Da li je DFS osobina garantovana?

Drugo asinhrono izvršenje daje za rezultat ovo stablo: niti BFS niti DFS

# A3 Pronalaženje DFS razapinjućeg stabla za zadati koren (1 / 2)

11

- kada koren izvodi prvi korak ili čvor (koji nije list) primi prvi  $M$ :
  - označi pošiljoca kao *predak* (ako nije koren)
  - za svakog suseda u seriji
    - pošalji mu  $M$
    - sačekaj prijem poruke "parent" ili „already" u odgovoru
  - pošalji poruku "parent" ka susedu *parent*
- kada procesor primi  $M$  inače (tj. nije slučaj od gore)
  - pošalji „already" ka pošiljaocu
  - koristi poruke "parent" i „already" za postavljanje promenljivih *children* i za odluku o završetku rada

# A3 Pronalaženje DFS razapinjućeg stabla za zadati koren (2/2)

12

- Ovaj algoritam garantuje da je razapinjuće stablo uvek DFS stablo.
- Analogan je sa sekvencijalnim DFS algoritmom.
- Složenost komunikacije:  $O(m)$  pošto se šalje konstantan broj poruka preko svakog luka
- Složenost obrade:  $O(m)$  pošto se istražuje svaki luk u seriji.

# A4 Pronalaženje razapinjućeg stabla bez zadatog korena

13

- Predpostaviti da procesori imaju jedinstvene identifikatore (inače nemoguće!)
- *Ideja:*
  - ▣ svaki procesor počinje izvršenje kopije algoritma za DFS razapinjuće stablo, postavljajući sebe kao koren stabla
  - ▣ označi svaku poruku sa ID inicijatora radi razlikovanja
  - ▣ kada se kopije „sudare“, kopija sa većim ID pobeđuje.
- Složenost komunikacije:  $O(nm)$
- Složenost obrade:  $O(m)$