Vaje 10

Thursday, December 13, 2018 9:39 AM

DISJUNKTNE MNOŽICE

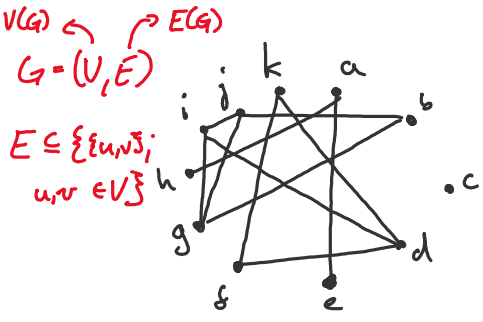
- ima univerzalno množico U
- zbirka disjunktnih množic $S = \{S_1, S_2 \dots S_k\}$, $S_i \subseteq U$
- vsaka množica S_i ima svojega predstavnika
- imamo tri osnovne operacije:
  - makeSet(x) ustvari eno-elementno ($x \in U$) množico $\{x\}$, katere predstavnik je x
  - union(S_i, S_j) navedi unijo $S_i \cup S_j$
  - FindSet(x) vrne predstavnika množice, ki vsebuje x ($x \in U$)

Povezane komponente

Vhod: neusmerjen graf G
Izhod: povezanostne komponente grafa

```
foreach v ∈ V(G) do
    makeSet(v)
end
for each {u,v} ∈ E(G) do
    if findSet(u) ≠ findSet(v) then union(u,v)
end
```

Simuliraj delovanje algoritma Povezanostne komponente na grafu G, kjer $V(G) = \{a, b \dots k\}$ in $E(G) = \{\{d,i\}, \{f,k\}, \{g,i\}, \{b,g\}, \{a,h\}, \{i,j\}, \{d,k\}, \{b,j\}, \{d,b\}, \{g,j\}, \{a,e\}\}$.



1. $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}, \{j\}, \{k\}$
$\{d,i\}$: $\{a\}, \{b\}, \{c\}, \{d,i\}, \{e\}, \{f\}, \{g\}, \{h\}, \{j\}, \{k\}$
$\{f,k\}$: $\{a\}, \{b\}, \{c\}, \{d,i\}, \{e\}, \{f,k\}, \{g\}, \{h\}, \{j\}$
$\{g,i\}$: $\{a\}, \{b\}, \{c\}, \{d,i,g\}, \{e\}, \{f,k\}, \{h\}, \{j\}$
$\{b,g\}$: $\{a\}, \{c\}, \{d,i,g,b\}, \{e\}, \{f,k\}, \{h\}, \{j\}$
$\{a,h\}$: $\{c\}, \{d,i,g,b\}, \{e\}, \{f,k\}, \{j\}, \{a,h\}$
$\{i,j\}$: $\{c\}, \{d,i,g,b,j\}, \{e\}, \{f,k\}, \{a,h\}$
$\{d,k\}$: $\{c\}, \{d,i,g,b,j,k,f\}, \{e\}, \{a,h\}$
$\{b,j\}$: $\{c\}, \{d,i,g,b\}, \{e\}, \{f,k,a\}, \{h\}, \{j\}$
$\{d,f\}$: $\{c\}, \{d,i,g,b\}, \{e\}, \{f,k,a\}, \{h\}, \{j\}$
$\{g,j\}$: $\{c\}, \{d,i,g,b\}, \{e\}, \{f,k,a\}, \{h\}, \{j\}$
$\{a,e\}$: $\{a,h,e\}, \{c\}, \{b,d,f,g,i,j,k\}$

(N) Naj bo G graf s k povezanostnimi komponentami. Če na G uporabimo Povezanostne komponente, kolikokrat sta poklicani operaciji findSet(x) in union(x,y) v odvisnosti od $|V(G)|, |E(G)|$ in k?

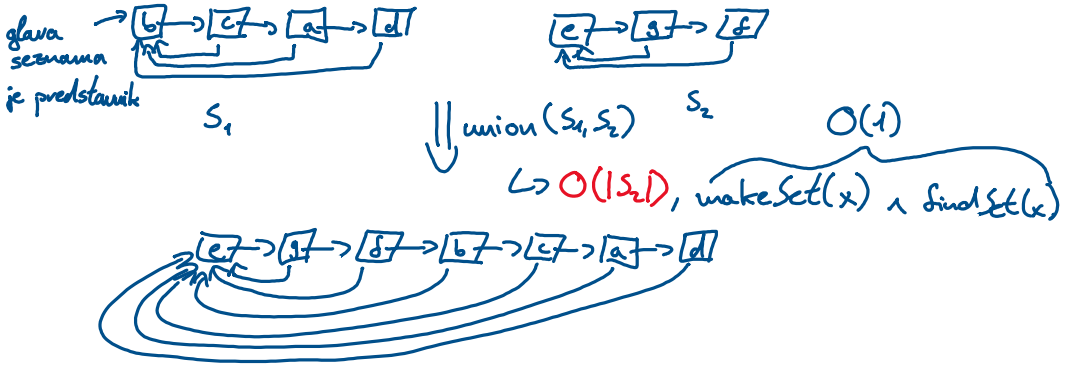
FindSet(x): $2|E(G)|$

union(x): $|V(G)| - 1 - (k-1) = |V(G)| - k$

// opazimo:
- na začetku imamo $|V(G)|$ eno-elem. množic.
- vsakič naredimo operacijo union(x,y) se število množic zmanjša za 1
- na koncu mora ostati k množic

Implementacija s povezanim seznamom

- množico predstavimo s povezanim seznamom



(N) Pokaži, da lahko zaporedje $n-1$ operacij union(x_i, x_j) na univerzalni množici z n elementi zahteva $\Theta(n^2)$

$U = \{x_1, x_2, \dots x_n\}$ za vsak $x_i \in U$ naredimo makeSet(x_i)

$\{\{x_1\}, \{x_2\} \dots \{x_n\}\}$

$\{\{x_1, x_2\}, \{x_3\} \dots \{x_n\}\}$

$\vdots$

union(x_2, x_1)
union(x_3, x_2)
$\vdots$
union(x_{i+1}, x_i)
union(x_n, x_{n-1})

$n-1$ operacij

$\sum_{i=1}^{n-1} O(i) = O(\sum_{i=1}^{n-1} i) = O(\frac{n(n-1)}{2}) = O(n^2)$

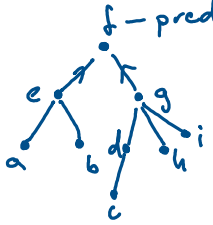
$\Rightarrow$ union$(\min\{|S_1|, |S_2|\})$

(N) Pokaži, da lahko zaporedje $n-1$ operacij union(x_i, x_j) na univerzalni množici z n elementi zahteva $O(n \log n)$.

Ideja: Kolikokrat se nekemu elementu x spremeni referenca na glavo?
- prvič se mu spremeni, ko kličemo union$(\{y\}, \{x\})$
  - opazimo, da po tej operaciji x pripada množici, ki ima vsaj 2 elementa
- naslednjič, ko se x-u spremeni referenca, združujemo z množico, ki ima, ki ima vsaj 2 elementa
- po tej operaciji bo x pripadal množici z vsaj štirimi elementi
- ko se elementu x naslednič spremeni referenca, združujemo z množico, ki ima vsaj 4 elemente.
- po tej operaciji bo imela množica, ki vsebuje x vsaj 8 elementov
- vidimo, da množica, ki vsebuje x, raste eksponentno
  - torej bo takšnih sprememb največ $O(\log(n))$

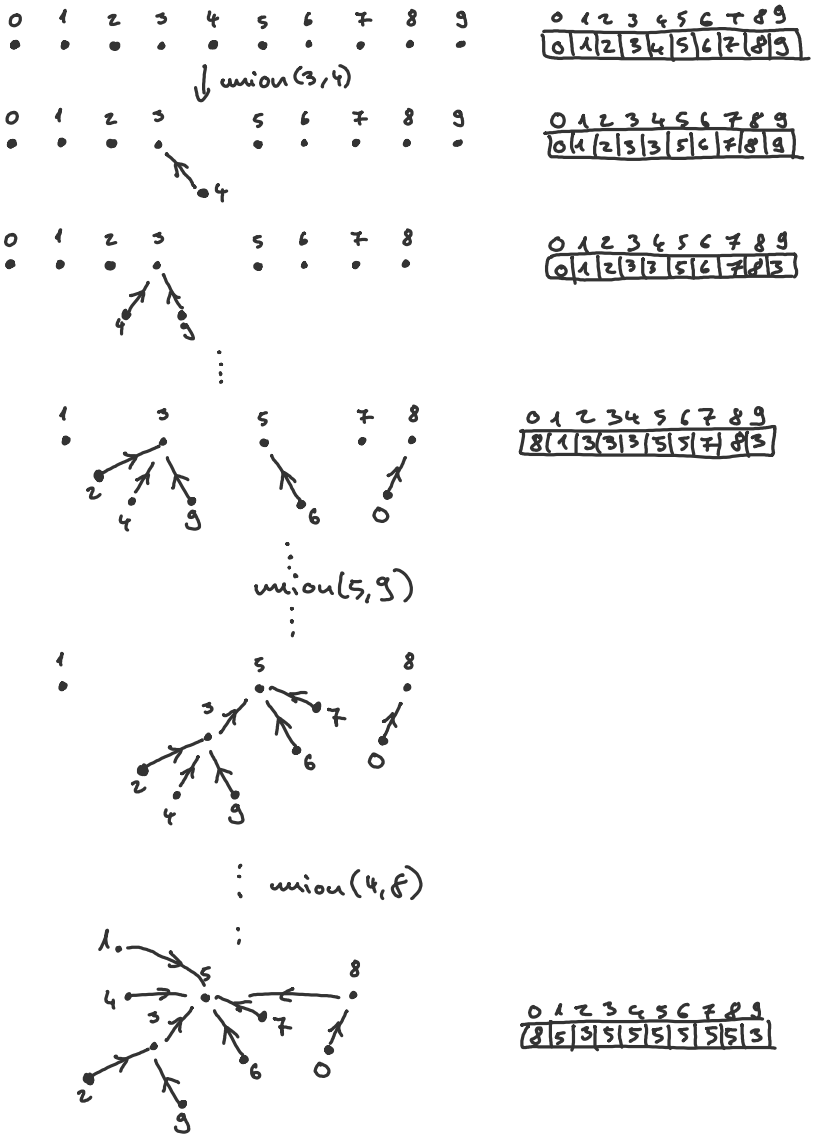
- Tovej vas $n-1$ operacij union stane $O(n \log n)$
  - ↳ torej $O(\log n)$ amortizirano na operacijo

Implementacija z drevesi



- makeSet(a)
  - a $O(1)$
- findSet(h)
  - sprehodimo se od vozlišče d do korena } O(najdaljša pot od lista do korena)
- union(x,y) $O(1)$
  - referenca enega korena nastavimo na drugi koren
- Heuristika 2: stiskanje poti
  - ↳ reference vozlišč na koren
- Heuristika 1: utežena unija:
  - drevo z manjšo višino pripnemo na drevo z večjo višino

(N) $U = \{0,1 \ldots 9\}$. Navedimo 10 operacij makeSet(i) za $i \in U$, nato pa union(3,4), union(4,9), union(8,0), union(2,3), union(5,6), union(5,9), union(7,3), union(4,8), union(6,1). Simuliraj z drevesi in obema heuristikama.



DN union
D na L

↳ $\begin{matrix} 3 & 0 \\ 5 & y \end{matrix}$ → stiskanje poti