Implementacija Dijkstrinog algoritma pomoću Fibonaccijeve hrpe

Jurica Horvat

Prirodoslovno-matematički fakultet, Zagreb

Siječanj, 2022.

Sadržaj

Uvod i motivacija

- Bavimo se problemom pronalaženja najkraćeg puta u grafu od jednog (fiksnog) čvora do svih ostalih
- Jedina pretpostavka na graf je da su **svi bridovi nenegativnih težina**
- Ponovimo kako Dijkstrin algoritam rješava ovaj problem

Uvod i motivacija

- Bavimo se problemom pronalaženja najkraćeg puta u grafu od jednog (fiksnog) čvora do svih ostalih
- Jedina pretpostavka na graf je da su **svi bridovi nenegativnih težina**
- Ponovimo kako Dijkstrin algoritam rješava ovaj problem

1. $G(V, E)$, s_0 izvorni čvor te skup $X = \emptyset$
2. $dist[], dist[x] = \infty, \ \forall x \in V \setminus \{s_0\}, \ dist[s_0] = 0$
3. Skup dostupnih čvorova X postepeno gradimo:
4. Uzmi element $y \in V \setminus X$ t.d. $dist[y] = \min_{z \in V \setminus X} dist[z]$
5. Izbaci y iz V i stavi ga u X
6. Za sve z t.d. je $\{y, z\} \in E$ popravljamo:
   $dist[z] = \min\{dist[z], dist[y] + l(y, z)\}$ pri čemu je $l(y, z)$ duljina brida od y do z

Uvod i motivacija

- Složenost Dijkstrinog algoritma s prioritetnim redom (binarna hrpa): $\mathcal{O}(|E| \lg |V| + |V| \lg |V|)$
- Za grafove s velikim brojem bridova imamo $lg|V|$ faktor uz $|E|$ kojeg želimo izbjeći
- S ovim će nam pomoći **Fibonaccijeva hrpa**
- Uvedimo najprije definiciju min-hrpe

Min-hrpa (eng. *Min-heap*)

Neka je H stablo. Kažemo da je H min-hrpa ako za svaki čvor $x \in H$ vrijedi $key(x) \leq key(y), \forall y \in S(x)$, pri čemu je $S(x)$ podstablo čvora x. Na skupu iz kojeg dolaze *key* vrijednosti čvorova mora biti definiran totalni uređaj.

- *U nastavku poistovjećujemo hrpu i min-hrpu zbog prirode problema kojim se bavimo*

Spojiva hrpa (eng. *Mergeable heap*)

Spojiva hrpa je struktura koja se sastoji od više usklađenih (min ili max) hrpi koje u parovima imaju prazan presjek s obzirom na nazive čvorova.

Spojiva hrpa podržava sljedeće operacije:

1. $MakeHeap()$
2. $Insert(H, x)$
3. $Minimum(H)$
4. $ExtractMin(H)$
5. $Union(H_1, H_2)$

Fibonaccijeva hrpa je poseban slučaj spojivih hrpi.

Ona podržava još i sljedeće operacije:

6. *DecreaseKey*(H, x, k)
7. *Delete*(H, x)

Usporedba složenosti s binarnom hrpom:

Procedure	Binary heap (worst-case)	Fibonacci heap (amortized)
MAKE-HEAP	$\Theta(1)$	$\Theta(1)$
INSERT	$\Theta(\lg n)$	$\Theta(1)$
MINIMUM	$\Theta(1)$	$\Theta(1)$
EXTRACT-MIN	$\Theta(\lg n)$	$O(\lg n)$
UNION	$\Theta(n)$	$\Theta(1)$
DECREASE-KEY	$\Theta(\lg n)$	$\Theta(1)$
DELETE	$\Theta(\lg n)$	$O(\lg n)$

Fibonaccijeva hrpa - *Insert*

```
FIB-HEAP-INSERT(H, x)
x.degree = 0
x.p = NIL
x.child = NIL
x.mark = FALSE
if H.min == NIL
    create a root list for H containing just x
    H.min = x
else insert x into H's root list
    if x.key < H.min.key
        H.min = x
H.n = H.n + 1
```

Složenost: $\mathcal{O}(1)$ - **lazy** pristup

Fibonaccijeva hrpa - *ExtractMin*

```
FIB-HEAP-EXTRACT-MIN(H)
z = H.min
if z ≠ NIL
    for each child x of z
        add x to the root list of H
        x.p = NIL
    remove z from the root list of H
    if z == z.right
        H.min = NIL
    else H.min = z.right
        CONSOLIDATE(H)
    H.n = H.n - 1
return z
```

Složenost: $\mathcal{O}(D(n))$

Fibonaccijeva hrpa - *ExtractMin*

```
FIB-HEAP-EXTRACT-MIN(H)
z = H.min
if z ≠ NIL
    for each child x of z
        add x to the root list of H
        x.p = NIL
    remove z from the root list of H
    if z == z.right
        H.min = NIL
    else H.min = z.right
        CONSOLIDATE(H)
    H.n = H.n − 1
return z
```

Složenost: $\mathcal{O}(D(n))$ - kasnije dokazujemo $D(n) \in \mathcal{O}(\lg n)$

Fibonaccijeva hrpa - *DecreaseKey*

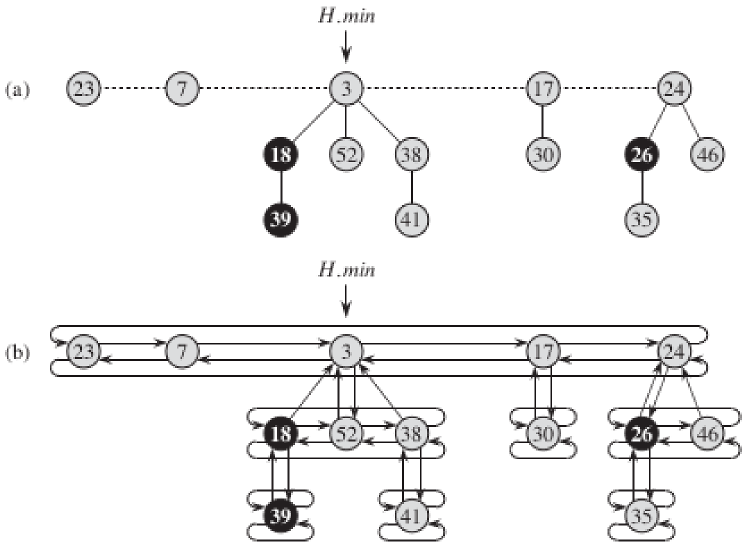
```
FIB-HEAP-DECREASE-KEY(H, x, k)
if k > x.key
    error "new key is greater than current key"
x.key = k
y = x.p
if y ≠ NIL and x.key < y.key
    CUT(H, x, y)
    CASCADING-CUT(H, y)
if x.key < H.min.key
    H.min = x
```

Složenost: $\mathcal{O}(1)$

Važni detalji implementacije:

- Struktura drži **head** pointer (*H.min* na slici dolje) na čvor s najmanjom key vrijednosti (u našem slučaju najmanja udaljenost do nekog čvora)
- Čvorovi u **root listi** su povezani cirkularnom dvostruko povezanom listom
- Sva djeca nekog čvora su također povezana cirkularnom dvostruko povezanom listom, a jedno dijete je povezano i s roditeljom

Fibonaccijeva hrpa



- Trebamo dokazati složenost $\mathcal{O}(lg|V|)$ operacija *ExtractMin* i *Delete*. Ovi rezultati će objasniti i ime strukture.

Lema 1

Neka je x proizvoljan čvor u Fibonaccijevoj hrpi i pretpostavimo da je $deg(x) = k$. Neka su $y_1, y_2, ..., y_k$ oznake djece čvora x poredane kronološki s obzirom na trenutak povezivanja s x: y_1 je najranije povezan s x, a y_k posljednji.
Tada je $deg(y_1) \geq 0$, $deg(y_i) \geq i - 2, \forall i \in \{2, 3, ..., k\}$.

Analiza složenosti

- Trebamo dokazati složenost $\mathcal{O}(lg|V|)$ operacija *ExtractMin* i *Delete*. Ovi rezultati će objasniti i ime strukture.

Lema 1

Neka je x proizvoljan čvor u Fibonaccijevoj hrpi i pretpostavimo da je $deg(x) = k$. Neka su $y_1, y_2, ..., y_k$ oznake djece čvora x poredane kronološki s obzirom na trenutak povezivanja s x: y_1 je najranije povezan s x, a y_k posljednji.
Tada je $deg(y_1) \geq 0$, $deg(y_i) \geq i - 2, \forall i \in \{2, 3, ..., k\}$.

- Dokaz: Očito je $deg(y_1) \geq 0$. Za $i \geq 2$ uočimo: u trenutku spajanja y_i na x imali smo $deg(x) \geq i - 1$ jer su tada $y_1, ..., y_{i-1}$ već spojeni s x. No, $deg(y_i) = deg(x)$ jer ih spajamo. Slijedi $deg(y_i) \geq i - 1$, a nakon spajanja je čvoru y_i odrezano najviše jedno dijete jer bi u protivnom odrezali i njega od čvora x. (vidjeti funkcije *Consolidate* i *Cut*)

Lema 2

Za sve $k \in \mathbb{N}_0$ vrijedi $F_{k+2} = 1 + \sum_{i=0}^{k} F_i$, pri čemu je F_k k-ti Fibonaccijev broj.

Lema 2

Za sve $k \in \mathbb{N}_0$ vrijedi $F_{k+2} = 1 + \sum_{i=0}^{k} F_i$, pri čemu je F_k k-ti Fibonaccijev broj.

- Dokaz: Indukcijom po k.

Lema 2

Za sve $k \in \mathbb{N}_0$ vrijedi $F_{k+2} = 1 + \sum_{i=0}^{k} F_i$, pri čemu je F_k k-ti Fibonaccijev broj.

- Dokaz: Indukcijom po k.

Lema 3

Za sve $k \in \mathbb{N}_0$ vrijedi: $F_{k+2} \geq \phi^k$, pri čemu je $\phi = (1 + \sqrt{5})/2$.

Lema 2

Za sve $k \in \mathbb{N}_0$ vrijedi $F_{k+2} = 1 + \sum_{i=0}^{k} F_i$, pri čemu je F_k k-ti Fibonaccijev broj.

- Dokaz: Indukcijom po k.

Lema 3

Za sve $k \in \mathbb{N}_0$ vrijedi: $F_{k+2} \geq \phi^k$, pri čemu je $\phi = (1 + \sqrt{5})/2$.

- Dokaz: Indukcijom po k. U koraku samo iskoristimo da je ϕ pozitivni korijen jednadžbe $x^2 = x + 1$.

Lema 4

Neka je x čvor u Fibonaccijevoj hrpi te $k = deg(x)$. Tada je $size(x) \geq F_{k+2} \geq \phi^k$, za $\phi = (1 + \sqrt{5})/2$, pri čemu je $size(x)$ veličina podstabla čvora x.

Dokaz:

Lema 4

Neka je x čvor u Fibonaccijevoj hrpi te $k = deg(x)$. Tada je $size(x) \geq F_{k+2} \geq \phi^k$, za $\phi = (1+\sqrt{5})/2$, pri čemu je $size(x)$ veličina podstabla čvora x.

Dokaz:

- Označimo sa s_k najmanju moguću veličinu podstabla nekog čvora x u proizvoljnoj Fibonaccijevoj hrpi, za kojeg vrijedi $deg(x) = k$.
- Jasno, $size(x) \geq s_k$ pa donju ogradu za $size(x)$ tražimo kao donju ogradu za s_k.
- Slijedi $size(x) \geq s_k \geq 2 + \sum_{i=2}^{k} s_{deg(y_i)}$, pri čemu su $y_1, \ldots, y_k$ djeca čvora x.
- Indukcijom pokazujemo $s_k \geq F_{k+2}$ te na kraju primijenimo *Lemu 3*.

Korolar 5

Za maksimalni stupanj nekog čvora u Fibonaccijevoj hrpi vrijedi: $D(n) \in \mathcal{O}(lg\, n)$, pri čemu je n broj čvorova u hrpi.

Dokaz:

Korolar 5

Za maksimalni stupanj nekog čvora u Fibonaccijevoj hrpi vrijedi: $D(n) \in \mathcal{O}(lg\, n)$, pri čemu je n broj čvorova u hrpi.

Dokaz:

- Neka je x proizvoljan čvor u Fibonaccijevoj hrpi s n čvorova te $k = deg(x)$.
- *Lema 4* povlači $n \geq size(x) \geq \phi^k$.
- Slijedi $k \leq log_{\phi} n$ i to dokazuje tvrdnju.

Analiza složenosti

Korolar 5

Za maksimalni stupanj nekog čvora u Fibonaccijevoj hrpi vrijedi: $D(n) \in \mathcal{O}(lg\,n)$, pri čemu je n broj čvorova u hrpi.

Dokaz:

- Neka je x proizvoljan čvor u Fibonaccijevoj hrpi s n čvorova te $k = deg(x)$.
- *Lema 4* povlači $n \geq size(x) \geq \phi^k$.
- Slijedi $k \leq log_\phi n$ i to dokazuje tvrdnju.

Ovim korolarom dokazali smo upitne složenosti izražene s $\mathcal{O}(D(n))$

Sada je ukupna složenost Dijkstrinog algoritma $\mathcal{O}(|E| + |V| \lg |V|)$

Testiranje

- Uspoređujemo efikasnosti Dijkstrinog algoritma s prioritetnim redom implementiranim pomoću Fibonaccijeve hrpe odnosno binarne hrpe.
- Primjeri *in*4, *in*5 su primjeri sa $|V| = 10^4$, $|E| = 10^6$, a *in*7 je primjer potpunog grafa sa $|V| = 4000$.

```
jurica@jurica:~/Desktop/oaaproj$ time ./fibodijk < in4 > out4

real    0m0,455s
user    0m0,439s
sys     0m0,016s
jurica@jurica:~/Desktop/oaaproj$ time ./dijkstra < in4 > out4

real    0m0,890s
user    0m0,854s
sys     0m0,037s
```

```
jurica@jurica:~/Desktop/oaaproj$ time ./fibodijk < in5 > out5

real	0m0,482s
user	0m0,465s
sys	0m0,016s
jurica@jurica:~/Desktop/oaaproj$ time ./dijkstra < in5 > out5

real	0m0,920s
user	0m0,866s
sys	0m0,041s
jurica@jurica:~/Desktop/oaaproj$
```

```
jurica@jurica:~/Desktop/oaaproj$ time ./fibodijk < in7 > out7

real	0m2,862s
user	0m2,774s
sys	0m0,088s
jurica@jurica:~/Desktop/oaaproj$ time ./dijkstra < in7 > out7

real	0m6,372s
user	0m6,237s
sys	0m0,125s
```

Literatura

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Intrdouction to algorithms, Third Edition, The MIT Press, 2009.