

Izbrana poglavja

**Fenwick tree, bipartite matching, lowest common ancestor,
heavy-light decomposition, centroid decomposition**

Priprave na računalniške olimpijade 2019/2020

Tomaž Hočevar

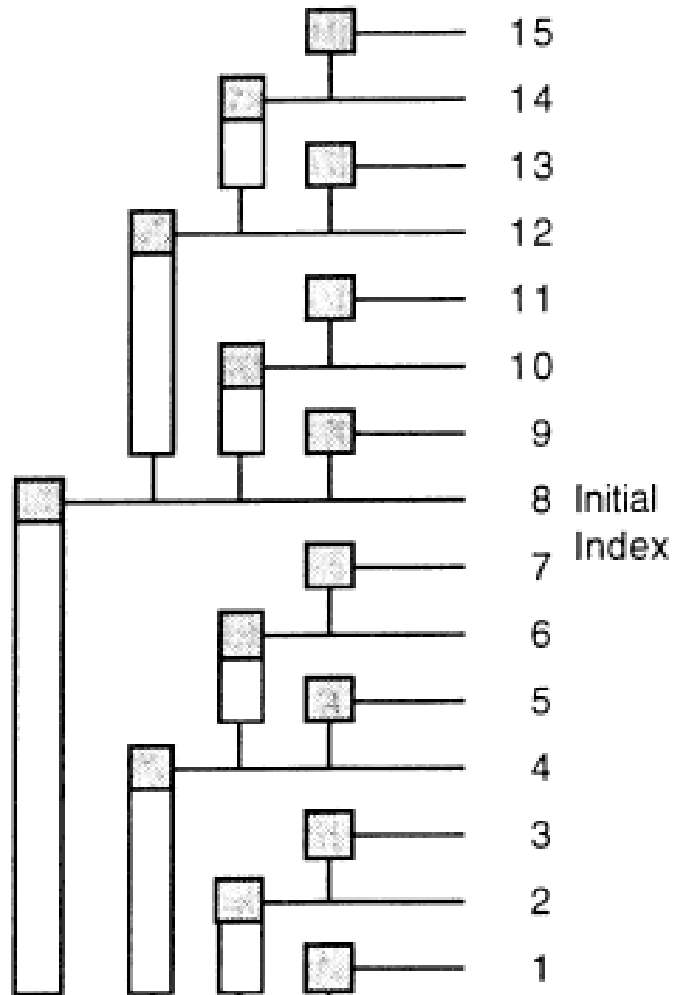
Fenwick tree

1	8	4	12	2	10	6	14	11	9	5	13	3	16	7	15
---	---	---	----	---	----	---	----	----	---	---	----	---	----	---	----

- `update(i, a) ... xi += a`
- `query(n) ... $\sum_{i=1..n} x_i = ?$`

Fenwick tree

CUMULATIVE FREQUENCY TABLES



Fenwick tree

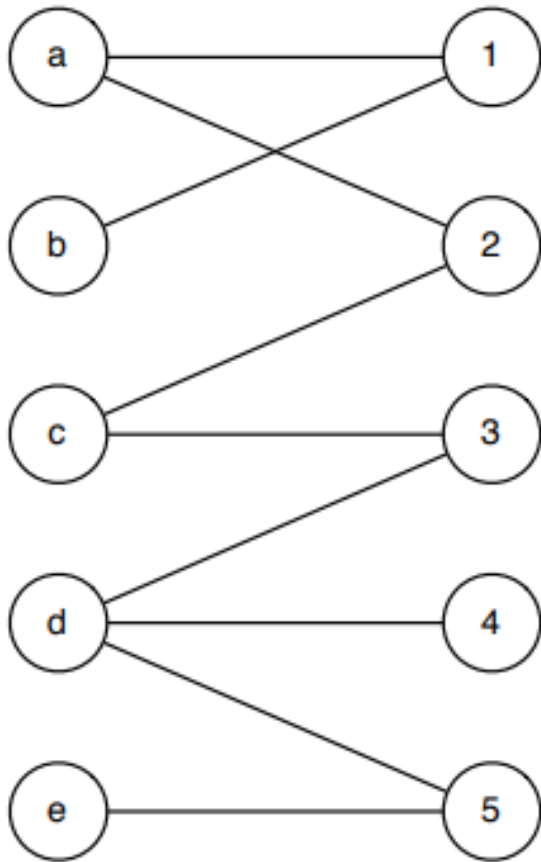
- binary indexed tree (BIT)
- predponoske poizvedbe
- kratka/učinkovita implementacija

```
#define N 100000
int BIT[N+1];

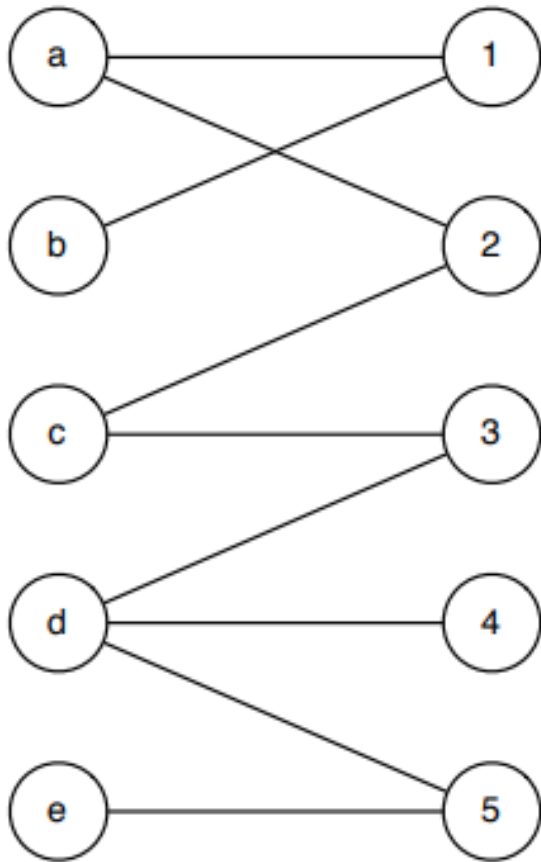
int query(int a) {
    int r=0;
    for (int x=a; x>=1; x-=x&-x) r+=BIT[x];
    return r;
}

void update(int a, int d) {
    for (int x=a; x<=N; x+=x&-x) BIT[x]+=d;
}
```

Bipartite matching

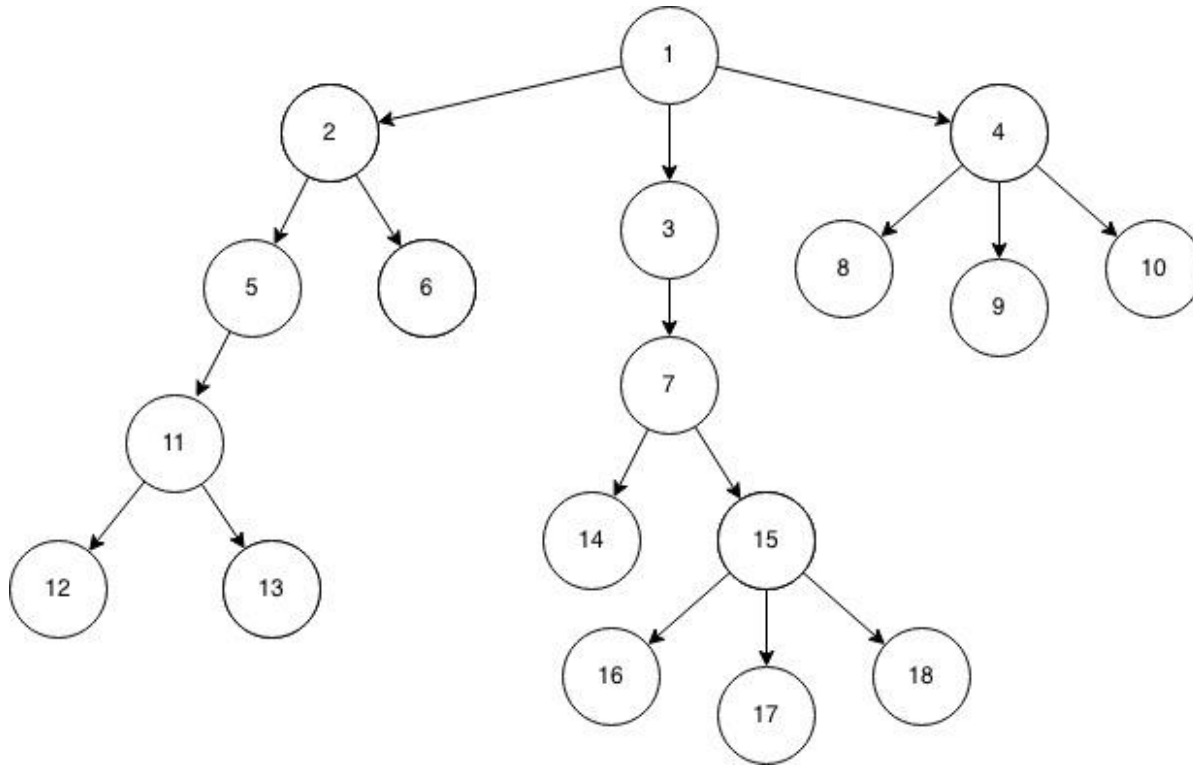


Bipartite matching

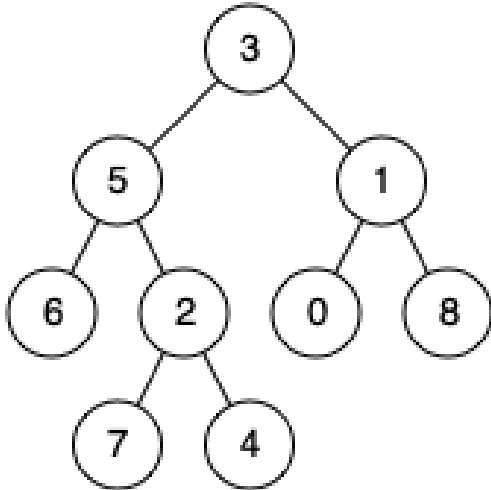


- prirejanje ... povezave nimajo skupnih vozlišč
- alternirajoča pot (alternating path)
- povečujoča pot (augmenting path)
- naj. prirejanje \equiv ni pov. poti
- implementacija
- $O(VE)$, $O(EV^{1/2})$
- pokritje, neodvisna množ.

Lowest common ancestor



Lowest common ancestor



1) LCA -> RMQ

2) predniki

– $a(x, k)$... 2^k -ti prednik x-a

$$a(x, k) = a(a(x, k-1), k-1)$$

– poravnamo globini

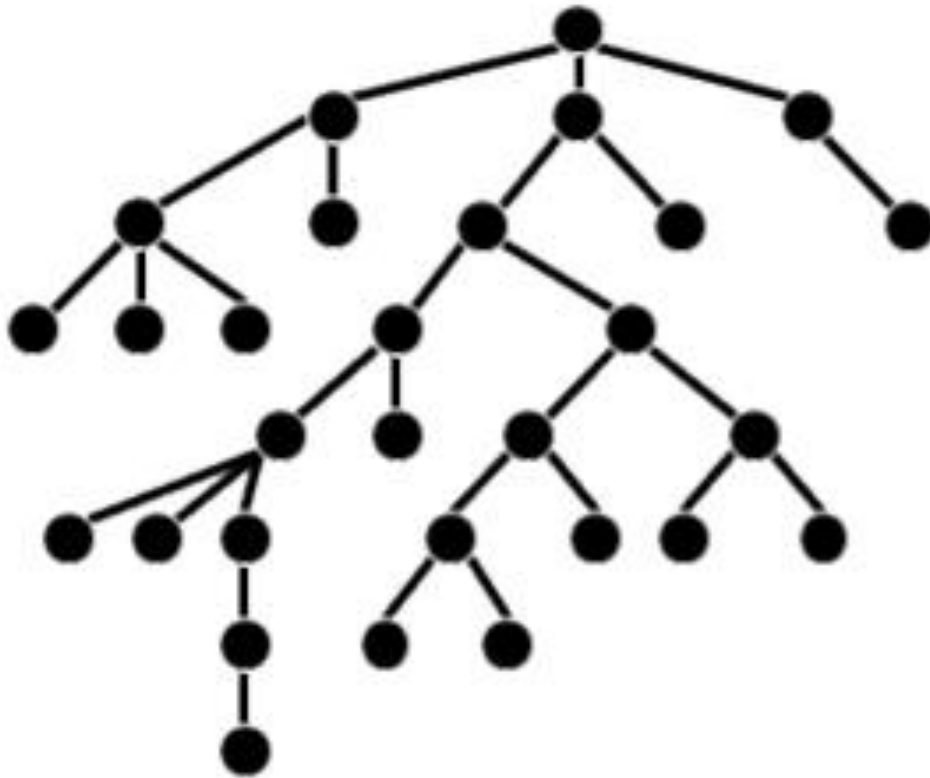
– skačemo pod najnižjega prednika

init = $O(n \log n)$

query = $O(\log n)$

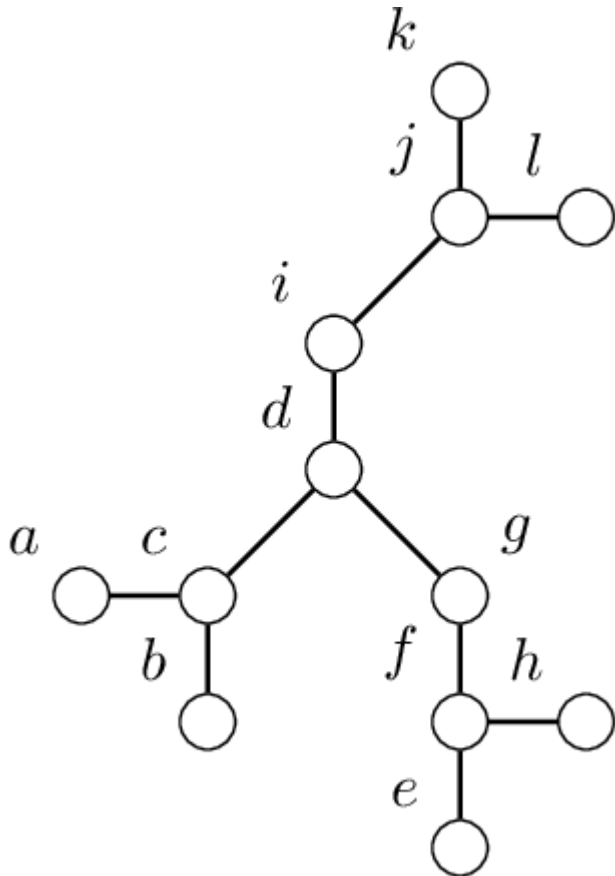
Heavy-light decomposition

- min. vozlišče na poti od A do B?



Centroid decomposition

- število poti dolžine K v drevesu z N vozlišči?



Centroid decomposition

- “deli in vladaj” na drevesu
- center \neq centroid
- iskanje centroida
- dekompozicija višine $O(\log n)$
- centroid ... “glavno” vozlišče na poti
- informacija o $O(n \log n)$ poteh

Centroid decomposition

- barvanje vozlišč, iskanje najbližjega pobarvanega

