ZADAĆA 8

Objektno orijentirano programiranje

Upute:

Zadaću je potrebno predati do 15. svibnja u 08:00 na Teamsu. Diskutiranje zadaće u okviru "study group" je dozvoljeno, ali predana zadaća mora biti samostalno riješena. Studenti koji predaju zadaću obavezni su prisustvovati vježbama, u suprotnome zadaća neće biti bodovana. Ako se kod ne prevodi (neovisno o tipu compiler-a), ili se događa greška prilikom izvršavanja koda, zadaća neće biti bodovana.

Zadatak 1. (40 bodova) Prioritetni red uz std::list

Implementirajte prioritetni red koristeći povezanu listu, a prema danoj deklaraciji:

```
template <typename K, typename V>
struct MinPriorityQueue {
    list<pair<K, V>> container;

    void insert(const pair<K, V>& el);
    V extractMin();

    // pronalazimo element s kljucem key, te ga postavljamo
    // na odgovarajuće mjesto s obzirom na njegov novi kljuc
    // newKey
    void decreaseKey(const K& key, const K& newKey);
    // ispis prioritetnog reda
    void print() const;


};
```

Instancirajte objekt klase `MinPriorityQueue` i redom unesite `{1, "osoba4"}`, `{2, "osoba7"}`, `{7, "osoba5"}`, `{8, "osoba1"}`, `{5, "osoba6"}`, a zatim ga ispišite koristeći `print` metodu. Zatim unesite novi `{6, "osoba9"}`, pa ispišite red, izvadite minimalni element pa ispišite red pa smanjite ključ s vrijednosti `6` na vrijednost `4`. Ispis treba biti:

K: 1, V: osoba4; K: 2, V: osoba7; K: 5, V: osoba6; K: 7, V: osoba5; K: 8, V: osoba1;

K: 1, V: osoba4; K: 2, V: osoba7; K: 5, V: osoba6; K: 6, V: osoba9; K: 7, V: osoba5; K: 8, V: osoba1;

K: 2, V: osoba7; K: 5, V: osoba6; K: 6, V: osoba9; K: 7, V: osoba5; K: 8, V: osoba1;

K: 2, V: osoba7; K: 4, V: osoba9; K: 5, V: osoba6; K: 7, V: osoba5; K: 8, V: osoba1;

Zadatak 2. (60 bodova) Minimalno-prioritetni red

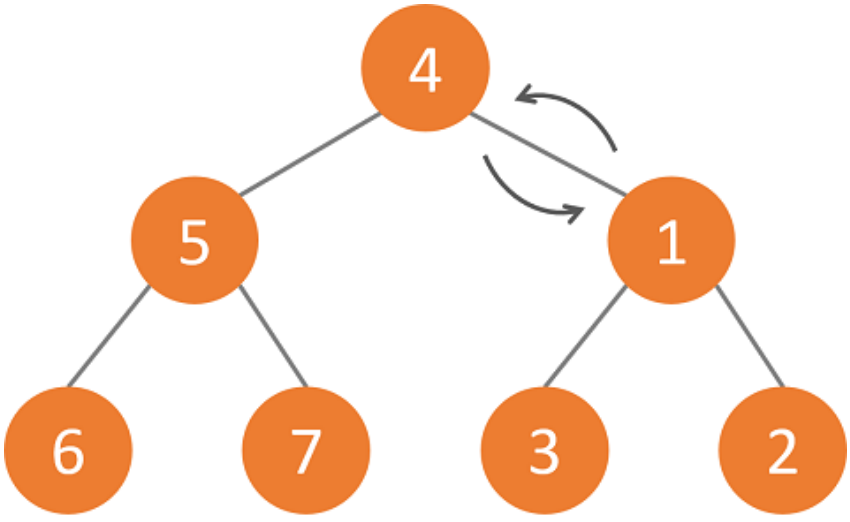
Kako je objašnjeno na predavanju, prioritetni red je struktura podataka koja efikasno održava poredak među čvorovima s obzirom na `key` vrijednost čvora.

Metode koje implementira minimalno-prioritetni red su:

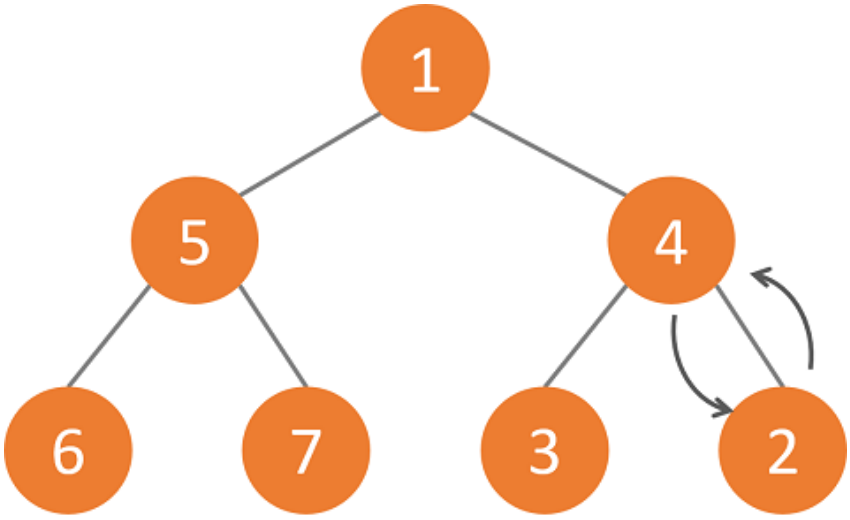
- `extractMin`
- `decreaseKey`
- `insert`

Kako bismo održavali poredak u prioritetnom redu, pozivamo metodu `downHeap` koja počevši od i-tog čvora s obzirom na `key` vrijednost čvora postavlja čvor na odgovarajuće mjesto. Npr. za poziv `downHeap(0)`, gdje `0` označava indeks korijena stabla, imamo sljedeće permutacije u hrpi:

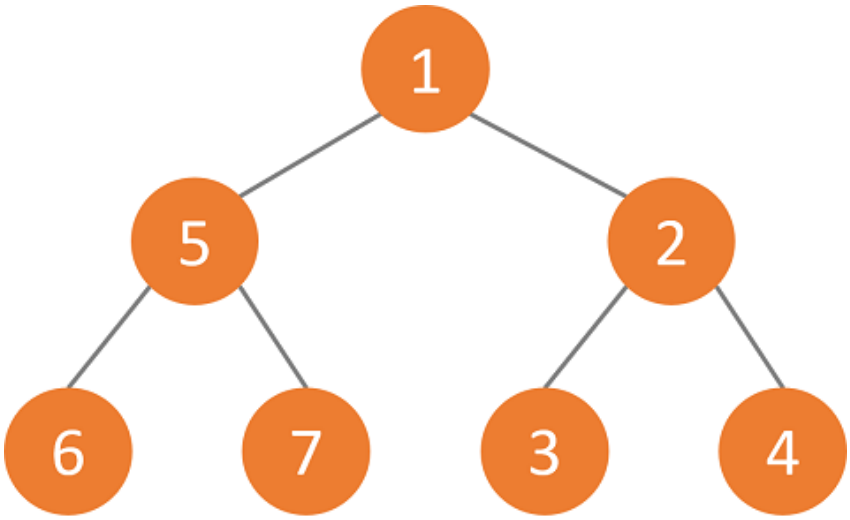
1)



2)



3)



Implementirajte prioritetni red s obzirom na sljedeću deklaraciju:

```
template <typename K, typename V>
class MinPriorityQueue {
public:
    CompleteBinaryTree<K, V> tree;
    MinPriorityQueue();
    MinPriorityQueue(const MinPriorityQueue&);

    void downHeap(Node i);
    void upHeap(Node i);

    // s obzirom na proslijeđeni vektor, izgrađujemo prioritetni red
    // (pretpostavljamo da je prioritetni red prazan, ili pregazimo
    // sve dosad pohranjene parove)
    void buildMinHeap(const vector<pair<K, V>>& L);

    V minimum() const;
    V extractMin();

    // pronalazimo element s indeksom i, te ga postavljamo
    // na odgovarajuće mjesto s obzirom na njegov novi kljuc
    // newKey
    void decreaseKey(Node i, K newKey);
    void insert(const pair<K, V>& v);
};
```

Instancirajte objekt klase `MinPriorityQueue` i redom unesite `{1, "osoba4"}`, `{2, "osoba7"}`, `{7, "osoba5"}`, `{8, "osoba1"}`, `{5, "osoba6"}`, a zatim ga ispišite uz pomoć operatora << pri čemu svaki par u redu također mora biti ispisan uz pomoć operatora `<<`. Zatim unesite novi `{6, "osoba9"}`, pa ispišite red, izvadite minimalni element pa ispišite red pa smanjite ključ ključ elementa s indeksom `2` na vrijednost `4` te ponovno ispišite. Ispis treba biti u sljedećem obliku (nije nužno da bude identičan nego da je poredak u prioritetnom redu održan):

K: 1, V: osoba4; K: 2, V: osoba7; K: 7, V: osoba5; K: 8, V: osoba1; K: 5, V: osoba6;

K: 1, V: osoba4; K: 2, V: osoba7; K: 6, V: osoba9; K: 8, V: osoba1; K: 5, V: osoba6; K: 7, V: osoba5;

K: 2, V: osoba7; K: 5, V: osoba6; K: 6, V: osoba9; K: 8, V: osoba1; K: 7, V: osoba5;

K: 2, V: osoba7; K: 5, V: osoba6; K: 4, V: osoba9; K: 8, V: osoba1; K: 7, V: osoba5;