Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Osnovne podatkovne strukture

Programiranje v višji prestavi

7. julij 2014

Tomaž Hočevar

Tekmovanja

- topcoder
- Codeforces
- CodeChef

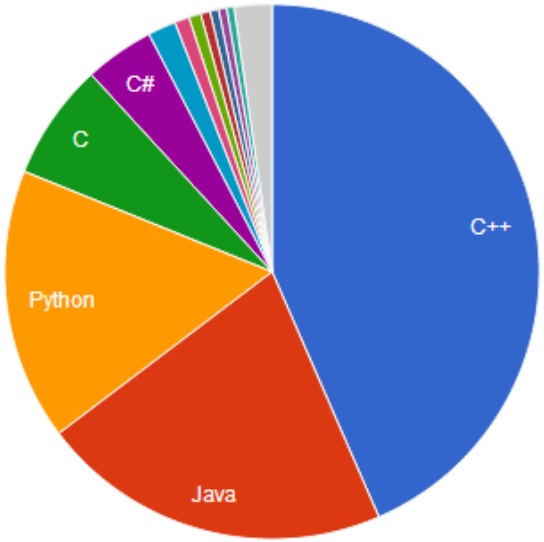
- Google Code Jam
- Facebook Hacker Cup

- IOI, CEOI
  - RTK (Tekmovanje ACM iz računalništva in informatike)
- ACM ICPC, CERC
  - UPM (Univerzitetni programerski maraton)

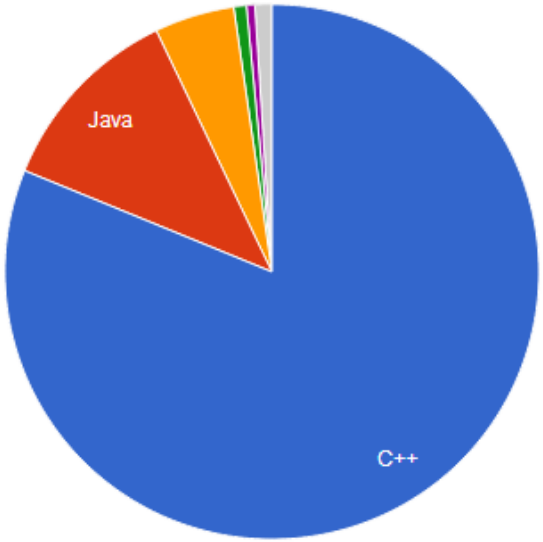
Programski jeziki

Codejam 2014 - [statistika]

Qualification (25000)



Round 3 (top 500)



C++ dokumentacija:

- http://www.cplusplus.com/reference/
- http://en.cppreference.com/

Osnovne podatkovne strukture

- tabela/polje (vector)
- seznam (list)
- vrsta (queue)
- sklad (stack)
- vrsta s prednostjo (priority queue)
- množica (set)
- slovar (map)

tabela/polje (vector)

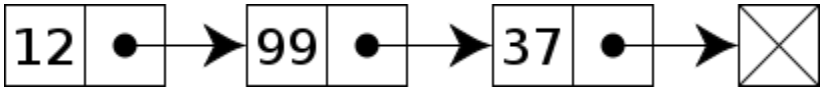
- dostop do vseh elementov
- velikosti so znane vnaprej
  - statična alokacija

```
#define N 100000
int tab[N];
```

- STL vector – 1D tabela dinamične velikosti

```
vector<int> v = {5,8,2};
v.push_back(3);
cout << v[v.size()-1] << endl;
```

seznam (list)



- insert, erase
- počasen dostop do elementa

- STL

```
list<int> l = {12,99,37};
l.insert(l.end(),4);
auto it = find(l.begin(),l.end(),99);
l.insert(it, 1);
l.erase(it);
for (int x : l) cout << x << endl;
```

- povezan seznam

vrsta (queue)

- FIFO – First In, First Out
- push, front, back, pop

- STL

```
queue<int> q;
q.push(9); q.push(1);
q.pop();
cout << q.front() << endl;
```

- tabela (krožna), povezan seznam

```
int q[N], h=0, t=0;
q[t++]=9; q[t++]=1;
h++;
cout << q[h] << endl;
```

primer uporabe – vector, queue

- dosegljivost v grafu (iz točke 0)

```
int n,e;
vector<int> sosedi[100000];
int mark[100000], st=0;

int main() {
    scanf("%d %d",&n,&e);
    for (int i=0;i<3;i++) {
        scanf("%d %d",&a,&b);
        sosedi[a].push_back(b);
        sosedi[b].push_back(a);
    }
    queue<int> q;
    q.push(0);
    mark[0]=1;
...
```

```
...
    while (!q.empty()) {
        int x=q.front();
        q.pop();
        st++;
        for (int y : sosedi[x])
            if (!mark[y]) {
                q.push(y);
                mark[y]=1;
            }
        }
    }
    cout << st << endl;
    return 0;
}
```

sklad (stack)

- FILO – First In, Last Out
- push, top, pop

- STL

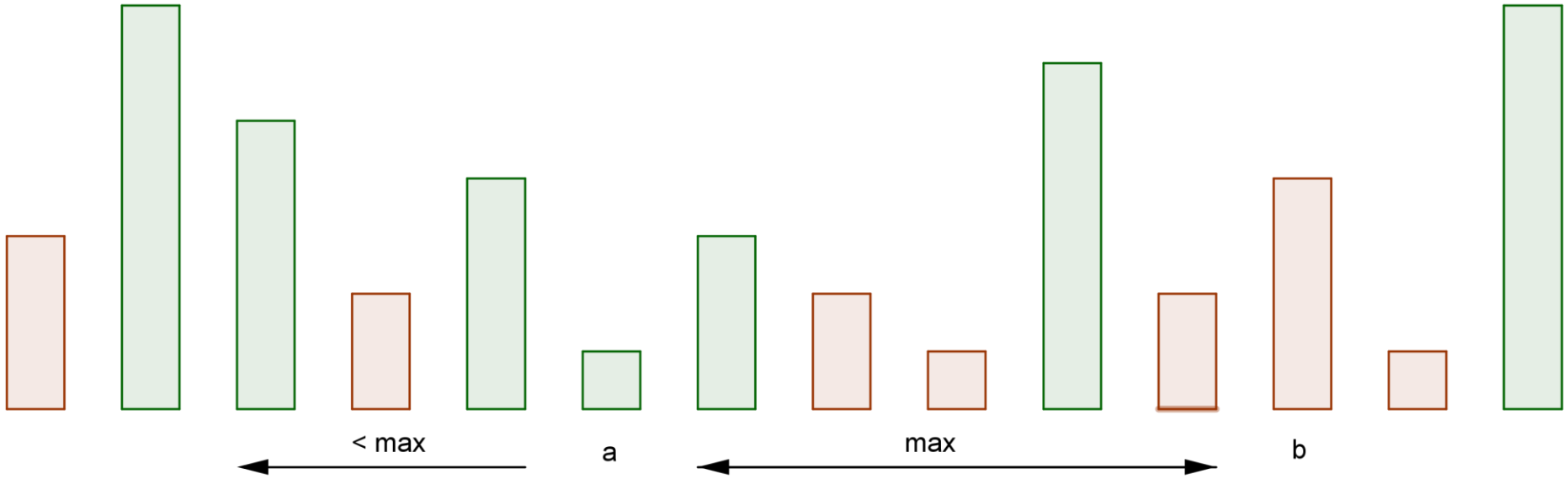
```
stack<int> s;
s.push(9); s.push(1);
s.pop();
cout << s.top() << endl;
```

- tabela, povezan seznam

```
int s[N], n=0;
s[n++]=9; s[n++]=1;
n--;
cout << s[--n] << endl;
```

sklad - primer

- rekurzija!
- CEOI 2014 – cake



vrsta s prednostjo (priority queue)

- vrsta s prioritetami elementov
  - najpomembnejši najprej

- STL

```
priority_queue<int> p;
p.push(1); p.push(9); p.push(3);
p.pop();
cout << p.top() << endl;
```

- seznam - slabo
- kopica (heap)
- drevesna struktura

vrsta s prednostjo - primer

- urejanje – heap sort
- iskanje najkrajše poti – Dijkstra

- naloga:

  Imamo n palic z dolžinami a_i, ki bi jih radi razrezali na čim manjše kose. Naredimo lahko k rezov – izberemo neko palico in jo na poljubnem mestu razrežemo na dva kosa. Tako dobimo n+1 palic, ostane pa nam še k-1 rezov. Želimo doseči, da bo na koncu najdaljša palica čim krajša! Kako?

  n=5, k=3

  a = [10,7,4,16,4]

množica (set)

- množica elementov – vsak je ali ni član množice
- insert, remove, contains, size, ...

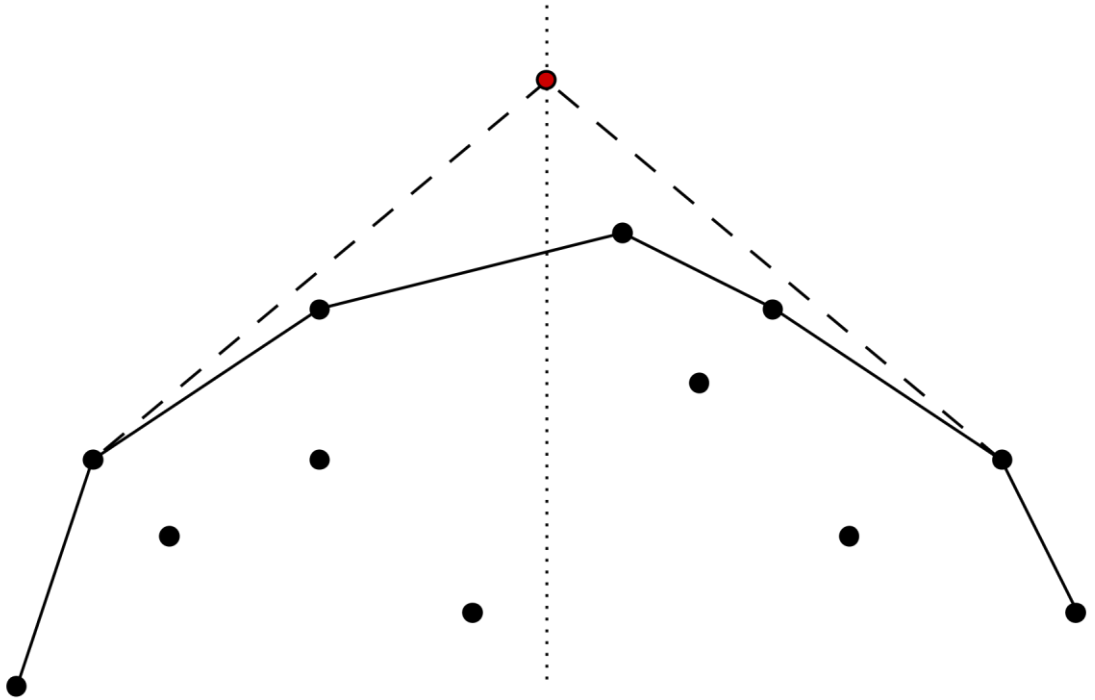
- STL

```
set<int> s = {6,8,2,8,4};
s.insert(6); s.insert(7);
s.erase(8);
cout << s.size() << " " << s.count(6);
```

  - implementiran z drevesno strukturo, kar lahko izkoristimo!
    - lower_bound (bisekcija)
    - iteratorji

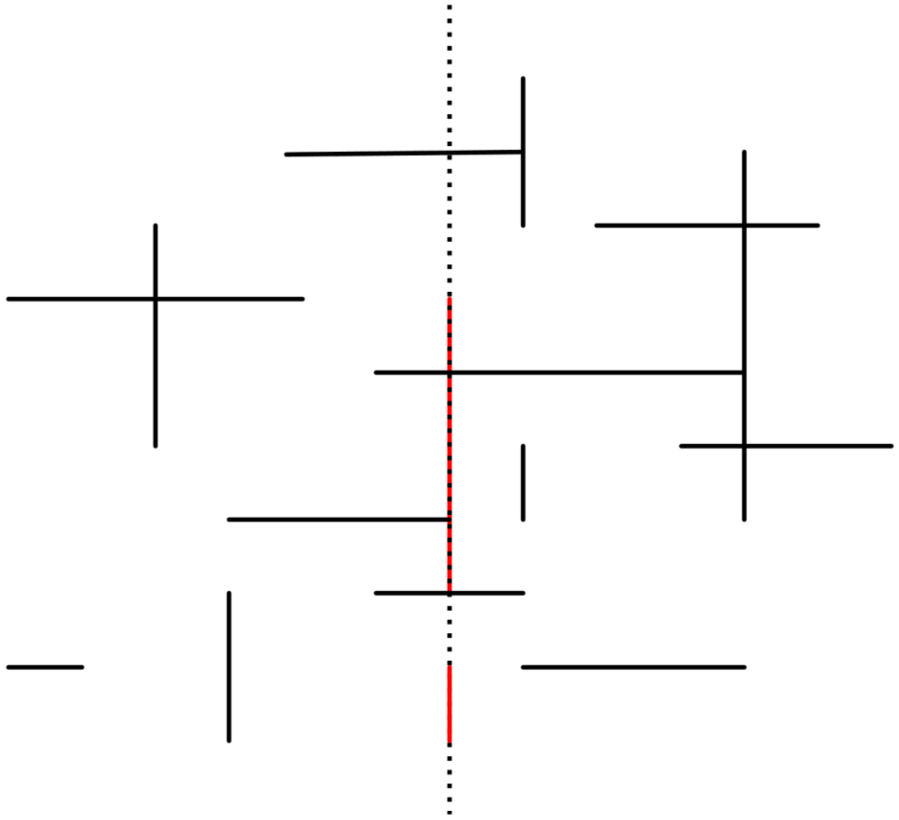
set – primer 1

- dinamična konveksna ovojnica
  - CEOI 2014 (practice session): surveyor



set – primer 2

- ali se kateri dve daljici iz množice vodoravnih in navpičnih daljic sekata?
  - sweep line, line segment intersection



slovar (map)

- pari <key, value>
- podoben set-u, vsak key se pojavi kvečjemu 1-krat

- STL

```
map<vector<int> ,int> m;
m[{0,0,1}]=1;
m[{1,1,0}]=6;
cout << m.count({1,0,1}) << endl;
```