

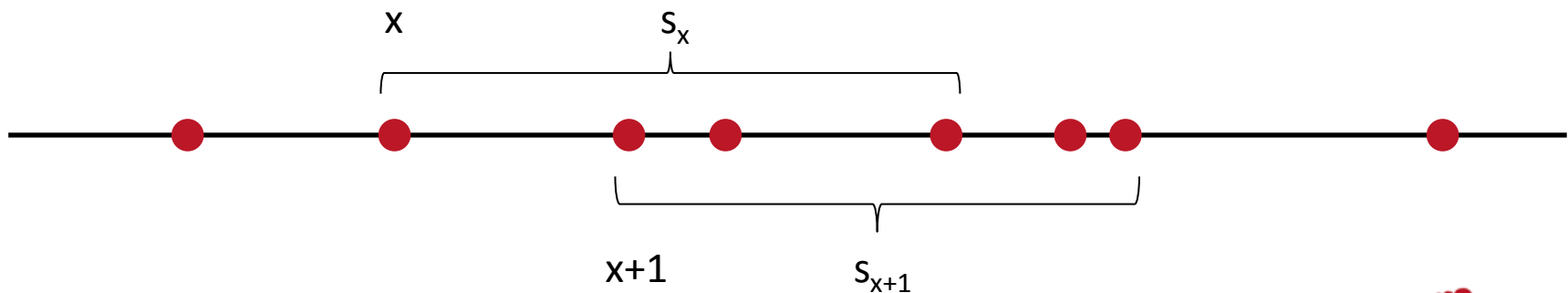
Univerzitetni programerski maraton

FINALE 2014 – rešitve nalog

Sosedski odnosi (//)

Preštej pare točk na premici, ki so na omejeni razdalji.

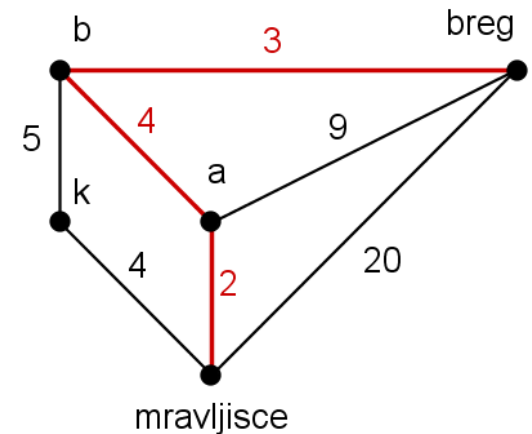
- $O(n^2)$ – TLE, 32-bitna števila – WA
- uredi točke
- sosednji točki imata podobne sosede
 - prehod z dvema kazalcema



Mravlje (/)

Izračunaj najkrajšo pot med dvema točkama v uteženem neusmerjenem grafu.

- Dijkstra, Bellman-Ford, Floyd-Warshall
- brute force
 - točk je samo 10
- branje, razčlenjevanje vrstice
- imena vozlišč
- vzporedne povezave, zanke
- mravljisce, breg nimata povezave



Floyd-Warshall

- dolžine najkrajših poti med vsemi pari točk v grafu

```
FOR (i,1,n) FOR (j,1,n) FOR (k,1,n) {  
    a[i][j] = min(a[i][j], a[i][k]+a[k][j]);  
}
```

Floyd-Warshall

- dolžine najkrajših poti med vsemi pari točk v grafu

```
FOR (i,1,n) FOR (j,1,n) FOR (k,1,n) {  
    a[i][j] = min(a[i][j], a[i][k]+a[k][j]);  
}
```



```
FOR (k,1,n) FOR (i,1,n) FOR (j,1,n) {  
    a[i][j] = min(a[i][j], a[i][k]+a[k][j]);  
}
```



Rimska zmešnjava (/)

Na koliko načinov lahko dani niz razbiješ na zaporedje rimskih števil < 4000?

DMXC → (D, M, X, C) (D, M, XC) (D, MX, C) (D, MXC)

- seznam rimskih števil
 - zapis enic, desetec, stotic, tisočic
 - najdaljše?

```
[ [' ', 'I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX'],  
  [' ', 'X', 'XX', 'XXX', 'XL', 'L', 'LX', 'LXX', 'LXXX', 'XC'],  
  [' ', 'C', 'CC', 'CCC', 'CD', 'D', 'DC', 'DCC', 'DCCC', 'CM'],  
  [' ', 'M', 'MM', 'MMM'] ]
```

Rimska zmešnjava (/)

- dinamično programiranje

CMDMCXCMXCMXMVIVIV**MXC**, MVI, VII, M, V, M

```
f[0]=1;
for (int i=1;i<=s.size();i++) {
    f[i]=0;
    for (int j=1;j<=20;j++) if (i-j>=0) {
        if (roman.count(s.substr(i-j,j))) {
            f[i]=(f[i]+f[i-j])%MOD;
        }
    }
}
```

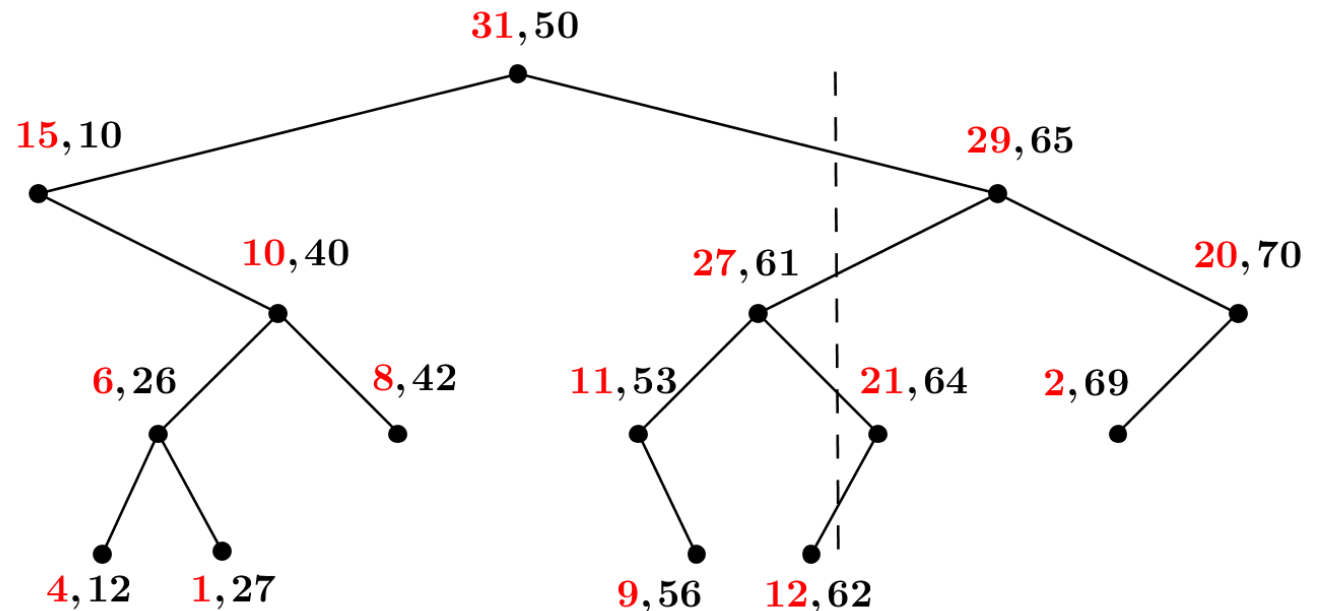
Izšteevanka (/)

Simuliraj dodajanje elementov in brisanje k-tega po velikosti.

- dvojiško iskalno drevo (uravnoreženo)
 - AVL, red-black, splay, ... , **treap**
 - težavna implementacija, rotacije
- “zakodirani” vhodni podatki
 - obdelava poizvedb po vrsti
 - prepreči vnaprejšnjo izgradnjo statičnega drevesa
- rebalance, fiksne delitvene točke

Treap

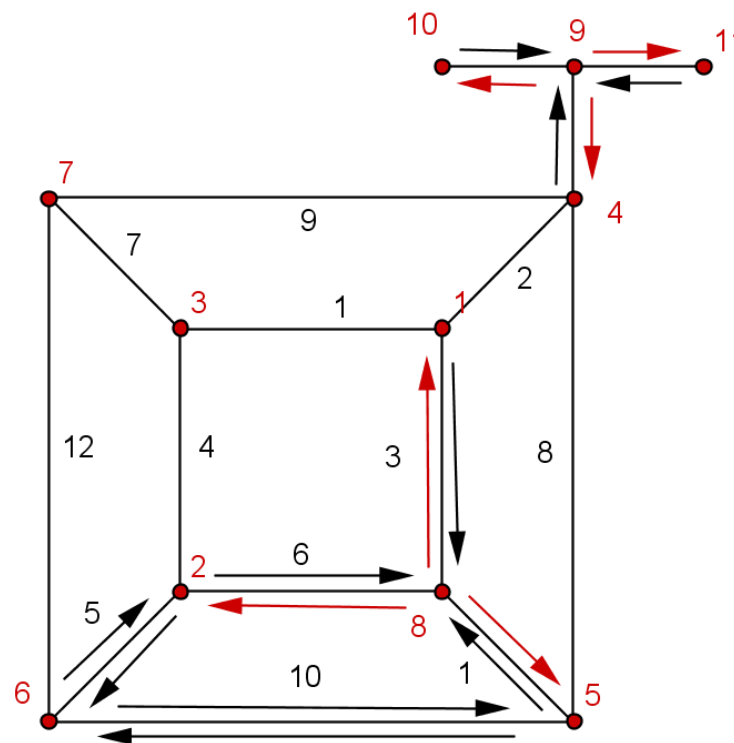
- tree + heap, naključno uravnoreženo drevo
- operacije: (prioriteta, vrednost)
 - split
 - merge
 - insert
 - delete
 - union



Izgubljena lica (/)

Poišči lica risbe povezanega ravninskega grafa, ki je narisano z ravnimi črtami.


- vsaka črta deli največ 2 lica
 - usmerjeni povezavi
- sledimo “sosednjim” povezavam → lica
- povezave iz vozlišča uredimo po kotu



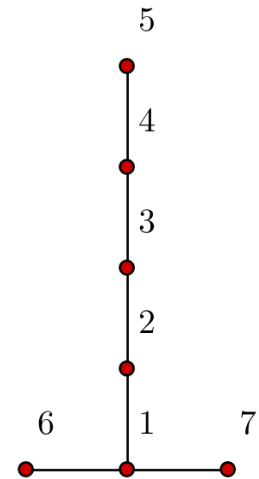
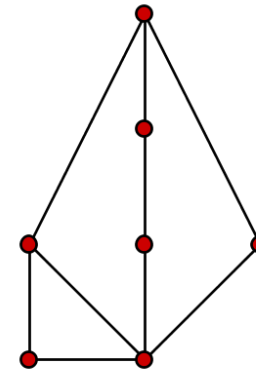
Kanonični opis

- minimalna ciklična rotacija zaporedja?
 - zanimiv problem

9 10 9 4 7 6 5 4 11

A horizontal sequence of numbers: 9, 10, 9, 4, 7, 6, 5, 4, 11. Above the last two numbers, '5' and '4', there is a horizontal arrow pointing to the left, with a vertical line extending upwards from the '4' and another vertical line extending downwards from the '5', connecting them to form a bracket-like shape.

- zaporedja niso splošna
 - opisi lic
 - zaporedni števili = povezava
 - 2 smeri
 - pazi drevesa!



- sortiranje kanoničnih opisov
 - povezava nastopa v največ 2 licih

Domače naloge (/)

Katerih k izmed n domačih nalog boš rešil najhitreje, če za nalogo x ob času t potrebuješ $a_x + t \cdot f_x$ časa, med nalogami pa potrebuješ odmor dolžine d ?

- znebimo se odmora: $a_x = a_x + d$
 - izbor nalog, vrstni red reševanja?
 - reševanje nalog po vrsti
 - dinamično programiranje
- $t(n,k)$... minimalen čas, da rešimo k izmed prvih n nalog
- $$t(n,k) = \min\{ t(n-1,k), t(n-1,k-1) + (a_n + t(n-1,k-1) \cdot f_n) \}$$

Domače naloge

- optimalen vrstni red reševanja nalog?
 - fiksni izbor nalog ($k=n$ oz. dobre vile)
 - zamenjava prvih dveh nalog?



$(a_x, f_x) (a_y, f_y) (a_z, f_z) \dots$? $(a_y, f_y) (a_x, f_x) (a_z, f_z) \dots$

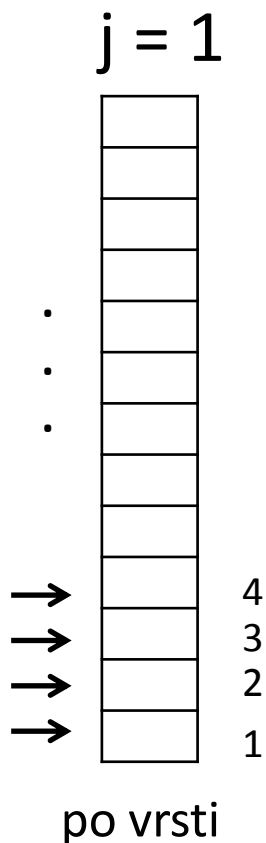
$$a_x + (a_y + a_x \cdot f_y) \leq a_y + (a_x + a_y \cdot f_x)$$

$$f_x / a_x \geq f_y / a_y$$

- vrstni red glede na padajočo vrednost f_x / a_x

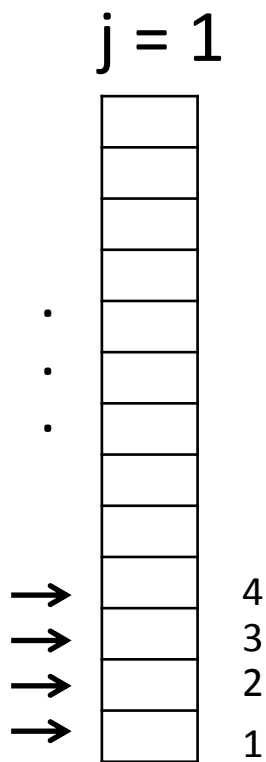
Eggstremna ornitologija (/)

Koliko metov potrebujemo, da ugotovimo, s katerega izmed n nadstropij se jajce še ne razbije, če imamo na voljo j jajc?

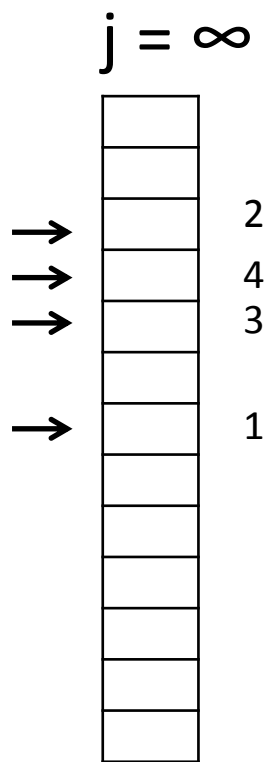


Eggstremna ornitologija (/)

Koliko metov potrebujemo, da ugotovimo, s katerega izmed n nadstropij se jajce še ne razbije, če imamo na voljo j jajc?



po vrsti

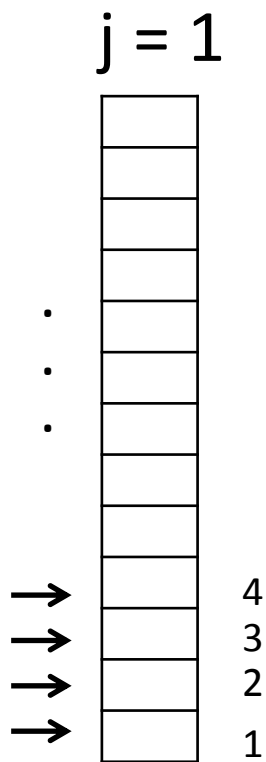


bisekcija

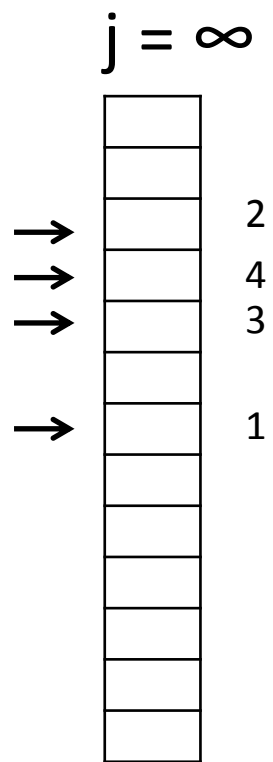
$j = 2$

Eggstremna ornitologija (/)

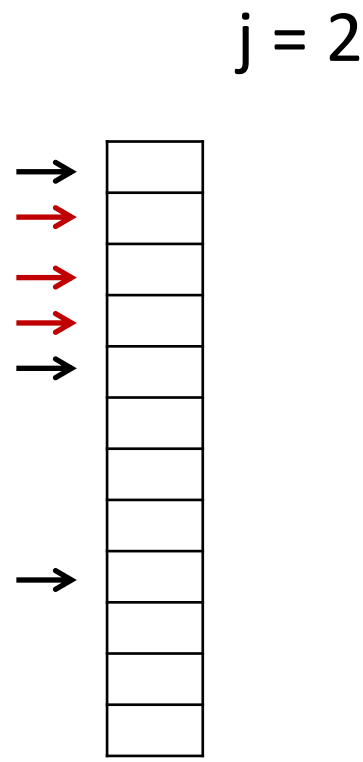
Koliko metov potrebujemo, da ugotovimo, s katerega izmed n nadstropij se jajce še ne razbije, če imamo na voljo j jajc?



po vrsti



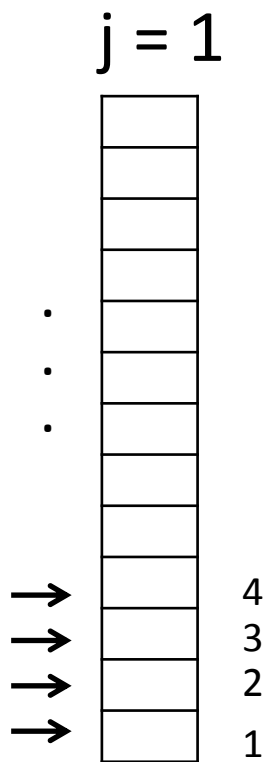
bisekcija



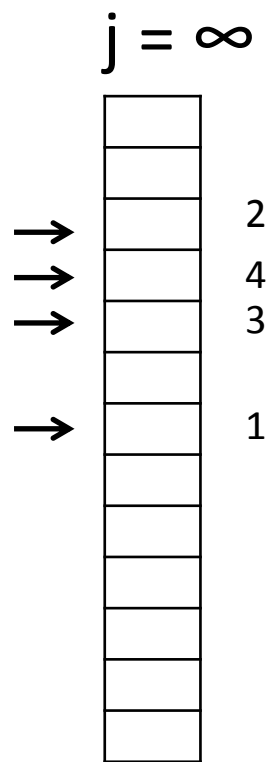
$m=6 (2\sqrt{n})$

Eggstremna ornitologija (/)

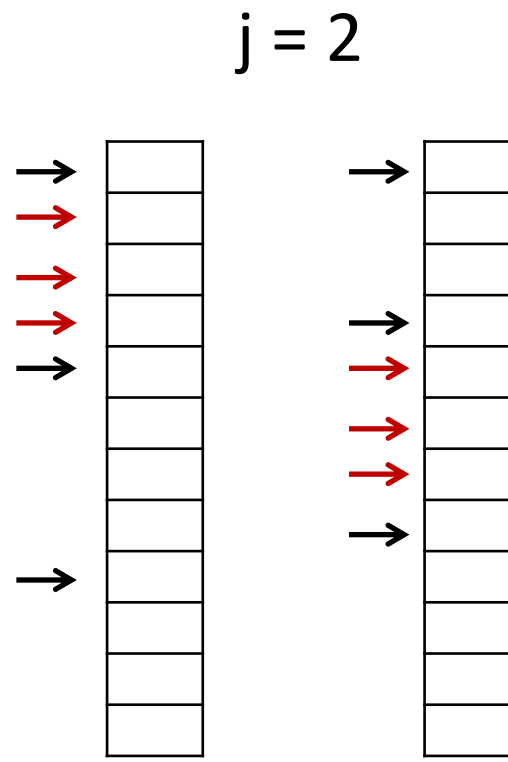
Koliko metov potrebujemo, da ugotovimo, s katerega izmed n nadstropij se jajce še ne razbije, če imamo na voljo j jajc?



po vrsti



bisekcija



$m=6$ ($2\sqrt{n}$)

$m=5$

Eggstremna ornitologija (/)

$$f(j,n) = ?$$

$$\dots = f(j,10^9) = f(j, 10^9+1) = \dots$$

$v(j,m)$... max. višina stavbe, ki jo lahko rešimo z j jajci in m metri.

$$v(j,m) = v(j-1,m-1) + 1 + v(j,m-1)$$

$$v(1,m) = m$$

$$v(2,m) = m-1 + 1 + v(2,m-1) = m \cdot (m+1) / 2$$

$$v(j,m) = O(m^j)$$

- tabeliramo za $j \geq 3$
- bisekcija

$$f(j,n) = \min. m: v(j,m) \geq n$$

Cenzura (/)

Preštej nize dolžine n , ki ne vsebujejo nobenega izmed prepovedanih podizov.

- en prepovedan podniz, npr. abaabb

$f(n, s_{n-5} \dots s_n)$ - število nizov, ki se končajo na $s_{n-5} \dots s_n$

$$f(n, \text{cccccab}) = f(n, \text{xayyyab})$$

$f(n, p)$ - število nizov, ki se končajo na predpono dolžine p

$$(n, 3) + 'a' \rightarrow (n+1, 4)$$

$$(n, 3) + 'b' \rightarrow (n+1, \mathbf{2})$$

– funkcija delnega ujemanja (failure function, KMP)

– $n \approx 10^9$

Potenciranje matrik

$(n, p) + \text{znak} \rightarrow (n+1, p')$

rekurenčna relacija, graf/avtomat

- število poti dolžine n , ki ne obišejo točke S

$$A \cdot f(n) = f(n+1), \quad A^n \cdot f(0) = f(n)$$

$$\begin{pmatrix} & & & & \\ & & & & \\ & & & & \\ z(p,p') & & & & \\ & & & & \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \begin{pmatrix} f(n,0) \\ \dots \\ f(n,S) \end{pmatrix}$$

- potenciranje matrike s kvadriranjem
- $O(\log n \cdot S^3)$

Zaključek

- dinamično programiranje
 - pogosta tema
 - “ne računaj istih/enakih stvari večkrat”
- geometrija
 - vaja
- knjige, zapiski
- predmet Tekmovalno programiranje