

# Univerzitetni programerski maraton

**FINALE 2015 – rešitve nalog**

Tomaž Hočevar

# Oglasna deska (17/19)

Simuliraj postavljanje in prekrivanje pravokotnikov na mreži.

- narišemo rob pravokotnika
- pobrišemo vse znotraj
- $O(n \cdot w \cdot h)$
  
- risanje pravokotnikov izven mize

```
10 5 4
3 3 4 3
8 3 10 10
-1 -1 3 4
-10 -10 2 3
```

```
|=====#
+                $
$ +---+ +---
$ |   |   |
# =+---+= |
```

# Robin Hood (14/20)

---

Na vsakem koraku zmanjšaj premoženje trenutno najpremožnejše osebe.

- kdo je trenutno najpremožnejši?
- $O(n^2)$  – TLE
- drevesna struktura
  - kopica, uravnoteženo drevo
- standardne knjižnice
  - `priority_queue`, `PriorityQueue`, `heapq`
  - `set`, `TreeSet`, ?

# Begunci (15/19)

Kje se nahaja neka oseba  $x$ , če smo jih po vrsti razvrstili v skupine danih velikosti?

$\min(i: \sum_{j \leq i} a_j \geq x)$

a) dvojiško iskanje / bisekcija

- delne vsote  $f_i = \sum_{j \leq i} a_j$
- $O(k \cdot \log(n))$

30



2	5	1	7	12	0	4	6	1	6	3	2
---	---	---	---	----	---	---	---	---	---	---	---

delne vsote

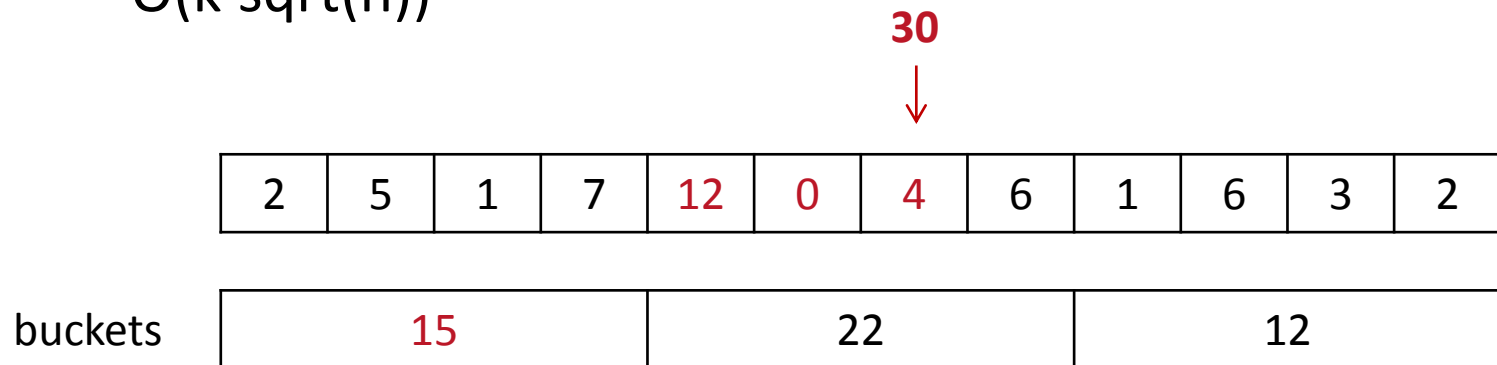
2	7	8	15	27	27	31	37	38	44	47	49
---	---	---	----	----	----	----	----	----	----	----	----

# Begunci

---

b) skupine po  $\sqrt{n}$  oseb

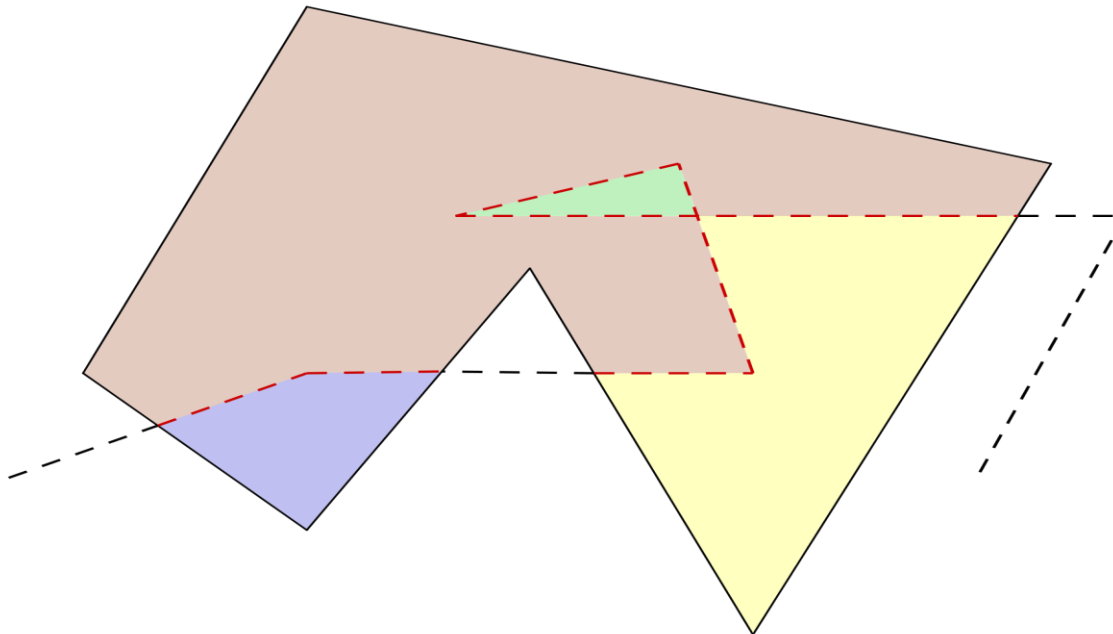
-  $O(k \cdot \sqrt{n})$



– število oseb > 32-bitno število

# Razrez (3/6)

Izračunaj vsoto obsega likov na katere razpade večkotnik, če ga razrežemo z lomljeno črto.



- obseg lika + 2·(dolžina črte znotraj lika)

# Presečišče daljic

- sta krajišči na različnih straneh?
- enačba premice skozi  $(x_1, y_1)$  in  $(x_2, y_2)$

$$A \cdot x + B \cdot y = C$$

$$A \cdot (x_1 - x_2) = B \cdot (y_2 - y_1)$$

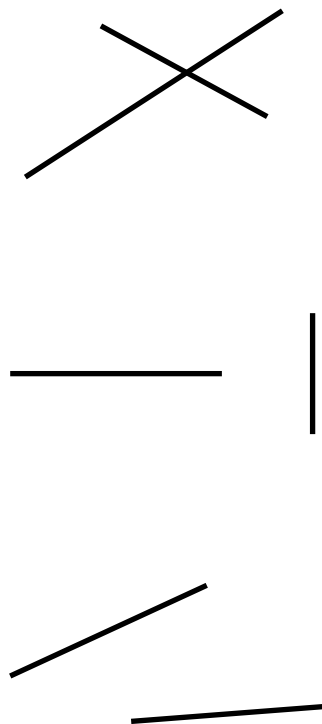
$$A = y_2 - y_1, \quad B = x_1 - x_2, \quad C = A \cdot x_1 + B \cdot x_2$$

- presečišče premic

$$\det = A_1 \cdot B_2 - A_2 \cdot B_1$$

$$x = (B_2 \cdot C_1 - B_1 \cdot C_2) / \det$$

$$y = (A_1 \cdot C_2 - A_2 \cdot C_1) / \det$$



# Trikotnik (2/7)

Na koliko načinov lahko podaljšaš palice, da bo z njimi mogoče oblikovati trikotnik.

- nemogoči trikotniki

$$\text{stranice } a \leq b \leq c \quad \rightarrow \quad a+b \leq c$$

- primeri, kjer je  $c$  najdaljša stranica (dolžine  $m$ )

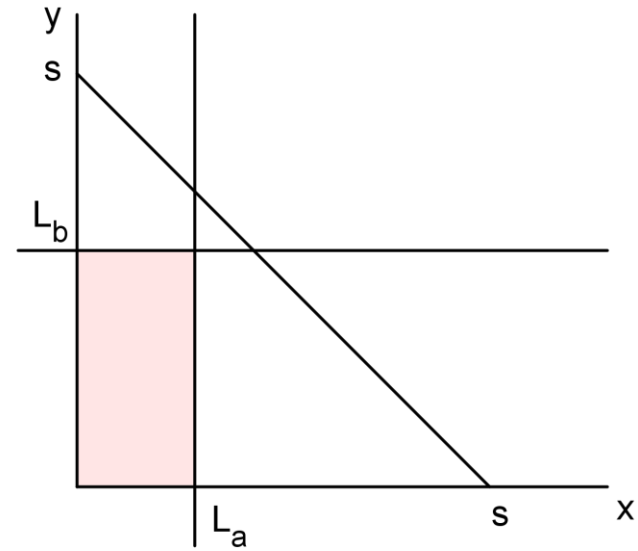
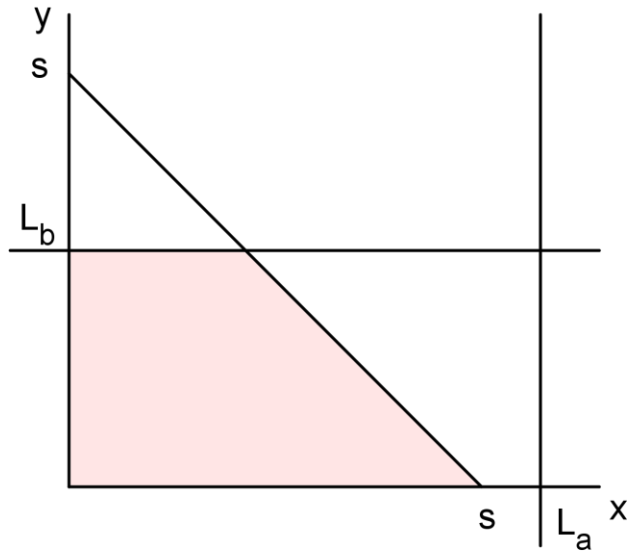
$$(a+x) + (b+y) \leq m \quad (\text{posledično } a+x < m, b+y < m)$$

$$0 \leq x \leq L_a, \quad 0 \leq y \leq L_b$$



# Trikotnik

$$x+y \leq m-a-b = s, \quad 0 \leq x \leq L_a, \quad 0 \leq y \leq L_b$$



test:    1 1 1  
          1 1 1

# Enaki nizi (0/8)

Nizom lahko permutiramo tretjine (rekurzivno). Ali sta podana niza enaka?

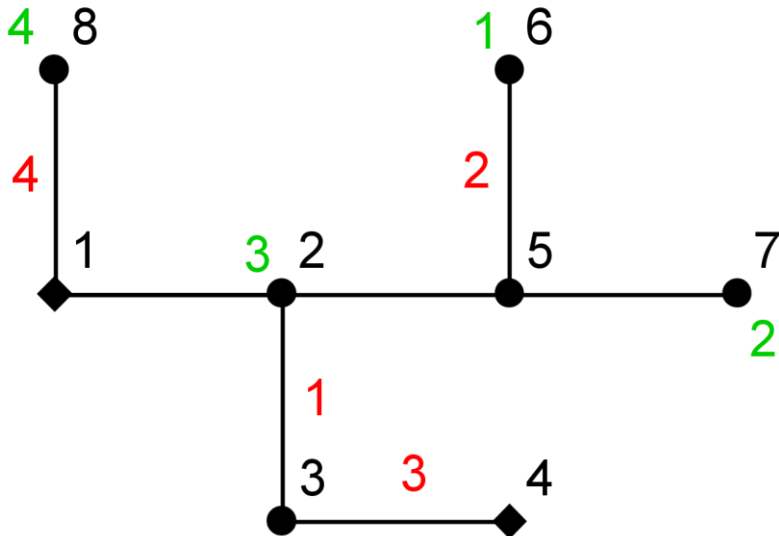
- normalna oblika
  - leksikografsko najmanjša varianta niza

```
def norm(s):  
    if len(s)==1: return s  
    if len(s)==3: return ''.join(sorted(s))  
    n = len(s)//3  
    t = sorted([norm(s[:n]), norm(s[n:2*n]), norm(s[2*n:3*n])])  
    return ''.join(t)
```

- $O(n \cdot \log(n))$

# Escape (1/2)

Čim hitreje pridi iz začetnega v končno vozlišče drevesa z zbiranjem ključev, ki odklepajo povezave.



	Z	1	2	3	4	K
Z		2			4	1,3
1	2		2	2	2,4	1,2,3
2		2			4	1,3
3		2			4	1,3
4	4	2,4	4	4		1,3,4
K	1,3	1,2,3	1,3	1,3	1,3,4	

- poti med ključi (Z, 2, 1, 3, K)

# Escape

---

- stanje: (trenutna lokacija, že zbrani ključi)
  - $k \cdot 2^k$
  - k možnih potez
- a) najkrajša pot
  - iskanje v širino (BFS)
- b) dinamično programiranje z memoizacijo
  - vedno dodamo nek nov ključ

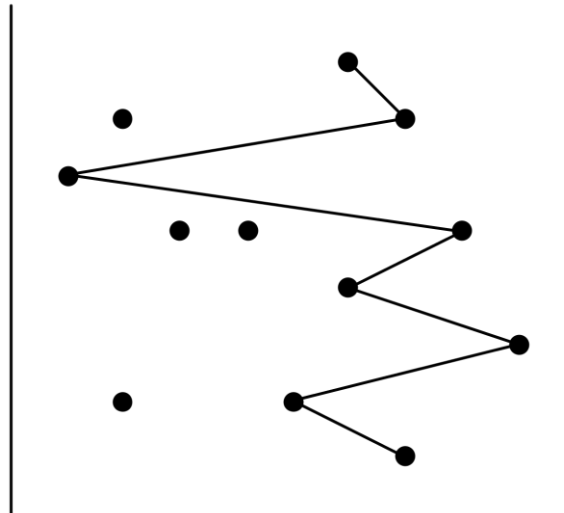
# Smučanje (1/5)

Obišči čim več količkov, pri čemer se ves čas premikaš navzdol in na vsakem obiskanem količku spremeniš smer.

$f(i, D)$  = optimalna rešitev od količka  $i$  proti desni

$$f(i, D) = \max_{j: y_j < y_i \wedge x_i < x_j} f(j, L) + 1$$

$O(n^2)$  - TLE



# Smučanje

---

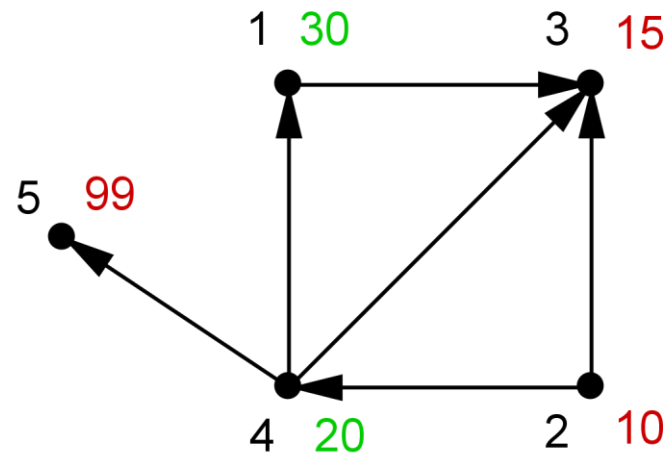
$$f(i, D) = \max_{j: y_j < y_i \wedge x_i < x_j} f(j, L) + 1$$

- podprobleme rešujemo po narašč.  $y$
- že rešene podprobleme hranimo v drevesni strukturi
  - trenutno opt. rešitev za vsako  $x$ -koordinato
  - Range Tree (Segment Tree)
  - Binary Indexed Tree (Fenwick Tree)
- $O(n \cdot \log(n))$

# Izguba časa (0/0)

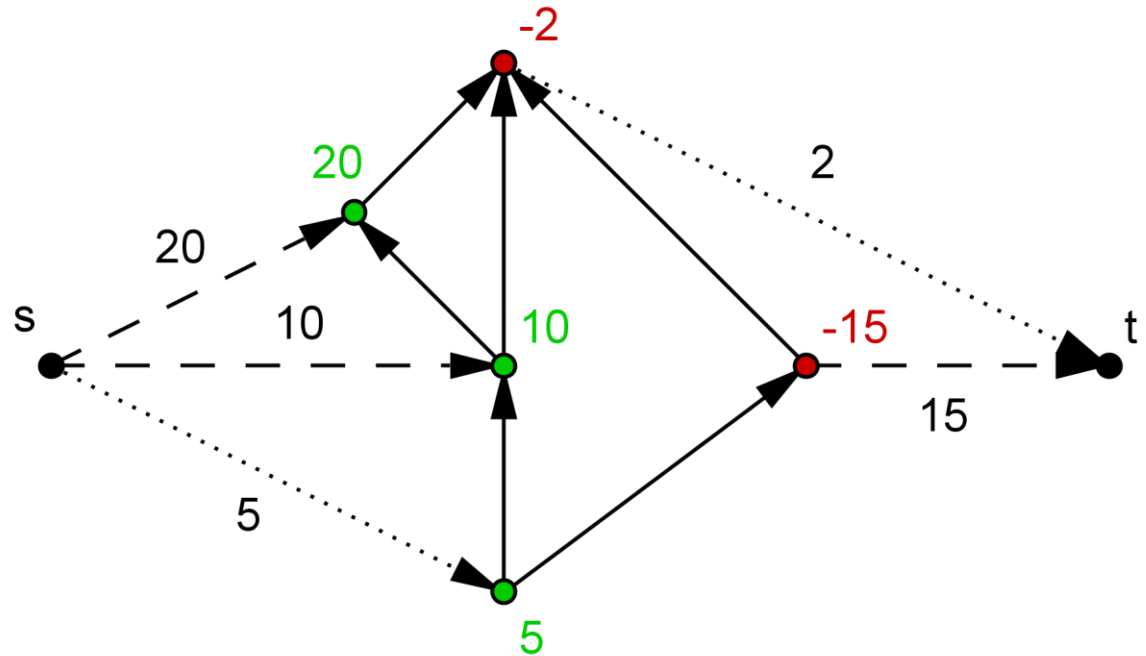
Optimalno popravi oz. izberi podmnožico nalog izmed že rešenih ter reši nekaj novih, da boš zadostil (pred)pogojem med nalogami.

```
5 6
30 10 15 20 99
1 0 0 1 0
3 1
4 2
3 2
1 4
3 4
5 4
```



# Closure problem

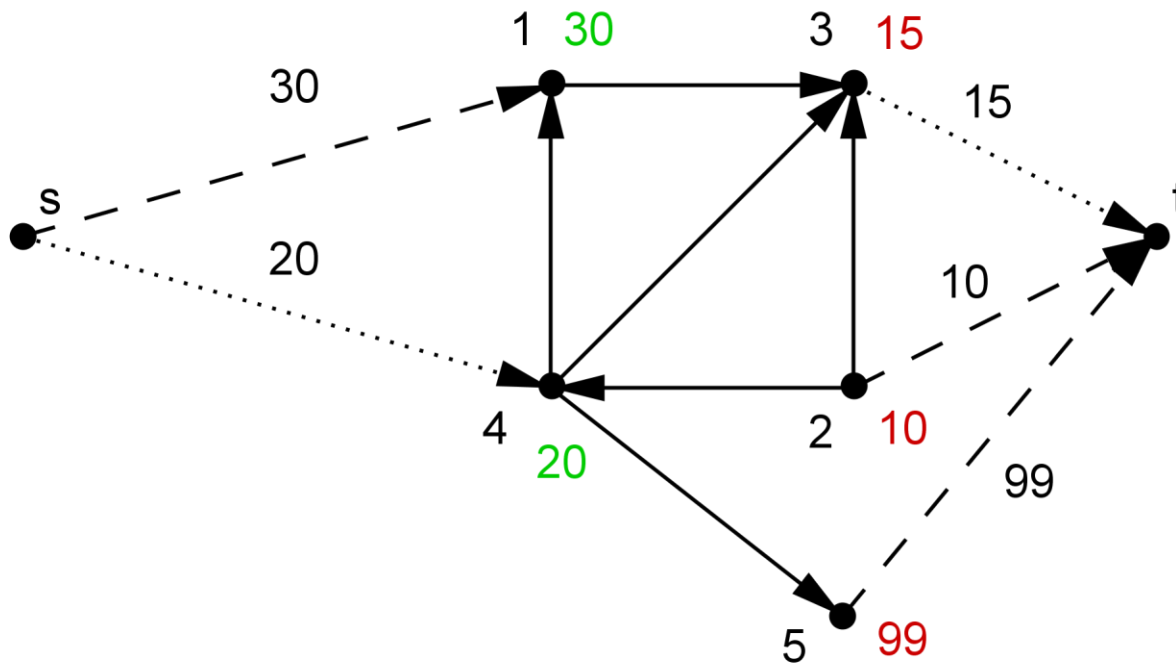
V usmerjenem grafu izberi množico vozlišč z največjo vsoto uteži, pri čemer se ne sme nobena povezava začeti v izbranem in končati v neizbranem vozlišču.



- minimum cut
- $\text{cena} = \text{vsi} - (\text{neizbrani} + \text{izbrani})$



# Izguba časa



# Tekmovalno programiranje

---

- izbirni predmet/dejavnost, 3 KT (UL FRI)
  - Računalništvo v praksi (1. stopnja)
  - Obštudijska strokovna dejavnost (2. stopnja)
- teme: dinamično progr., teorija grafov, rač. geom., ...
- tedensko reševanje nalog na Codeforces
- uvodni sestanek
  - prijave: [tomaz.hocevar@fri.uni-lj.si](mailto:tomaz.hocevar@fri.uni-lj.si)