

# Univerzitetni programerski maraton

**FINALE 2021 – rešitve nalog**

Tomaž Hočevar

# Delitev zemljišča

Tabelo želimo z vodoravnim in navpičnim rezom razdeliti na čim bolj podobna območja.

- minimiziramo razliko med vsoto največjega in najmanjšega kosa
- $O(w \cdot h)$  različnih delitev
  - $O(w \cdot h)$  operacij za seštevanje vrednosti v kosih je prepočasi
- a) pospešimo z 2D kumulativnimi vsotami
- b) zaporedoma obravnavamo podobne delitve npr.  $(x,y)$ ,  $(x+1,y)$ 
  - 1D kumulativne vsote

```
4 | 5 2 7
9 | 3 1 0
--+-----
8 | 7 4 5
```

# Ničle polinoma

Poišči koeficiente polinoma z ničlami  $a, a+1, \dots, b$ .

- $p(x) = (x-a)(x-(a+1))\dots(x-b)$
- vsak nov člen pomnožimo z že izračunanim polinomom  
 $(ax^3 + bx^2 + cx + d)(x-k)$   
 $(ax^4 + bx^3 + cx^2 + dx) - (kax^3 + kbx^2 + kcx + kd)$   
 $ax^4 + (b-ka)x^3 + (c-kb)x^2 + (d-kc)x + (-kd)$
- $O(n^2)$ , pazi modulo:  $-4 \% 3 = -1$  ali  $2$

```
p = [1]
for n in range(a,b+1):
    p.append(p[-1])
    for i in reversed(range(len(p)-1)):
        p[i] = (p[i]*(-n) + (p[i-1] if i>0 else 0))
        p[i] %= 1000000007
```

# Varnost na letališču

---

V besedilu iščemo prepovedane fraze.

- besedilo združimo v dolg niz
- fraze in besedilo razbijemo na zaporedje besed
  - znake, ki ne pašejo k besedam [ $\wedge$ a-zA-Z0-9] zamenjamo s ' ' in razrežemo (split)

V kovcku je  
se vec oblek, ena baterija in voda, ki se  
je rahlo razlila zaradi grdega ravnanja letaliskega osebja.

['V', 'kovcku', 'je', 'se', 'vec', 'oblek', 'ena', 'baterija', ...]

- iščemo pojavitve zaporedja besed fraze v besedah besedila
  - malo podatkov: brute-force

# Predmetni moduli

---

Iz vsakega modula želimo izbrati en predmet, da bo vsota KT čim večja, ne bo pa presegla meje  $k$ .

- KT so cela števila, meja  $k \leq 1000$ .
- podobno problemu nahrbtnika
- $f(m,k) = \text{true/false}$  ... ali lahko iz prvih  $m$  modulov izberemo predmete z vsoto kreditnih točk natanko  $k$ 
  - kateri predmet v  $m$ -tem modulu bomo vzeli?
- $O(mkp)$  časa,  $O(mk)$  prostora

# Vzporedno izvajanje

Funkcije ob vzporednem izvajanju spreminjajo isti globalni števec. Kakšna je najmanjša možna končna vrednost?

- $i$ -ta funkcija se mora izvesti  $A_i$ -krat
  - na začetku prebere števec, na koncu zapiše za ena povečano vrednost

```
function F_i
  for j = 1 .. K_i
    f_i()
```

```
function f_i
  cnt = read()      // iR
  cnt = cnt + 1
  ...
  write(cnt)       // iW
```

- lahko dosežemo  $\min(A_i)$ , recimo da je to  $m$ -ta funkcija
  - $mR(0)$  [vse ostalo]  $mW(1)$   $mR(1)$   $mW(2)$  ...  $mR(A_m-1)$   $mW(A_m)$
  - lahko pa tudi bolje

# Vzporedno izvajanje

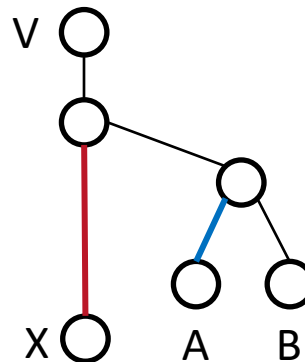
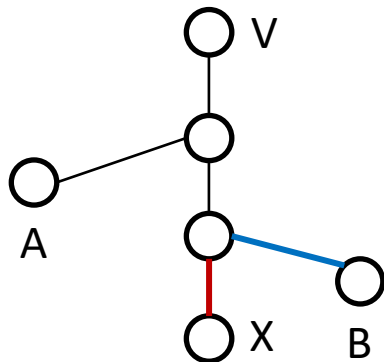
---

- $N=1: A_1$
- $N>1:$ 
  - $\forall A_i = 0 \rightarrow 0$
  - $\exists A_i = 1 \rightarrow 1 \quad iR(0) \text{ [vse ostalo] } iW(1)$
- $N \geq 2, \forall A_i \geq 2$ 
  - rezultat bo vsaj 2:  $\dots iR(?) \dots iW(?) \dots iR(\geq 1) \dots iW(\geq 2)$
  - rezultat je lahko 2:
    - $aR \ aW \ aR \dots aW \ aR \text{ [vse ostalo] } aW$
    - bi lahko “resetirali” števec pred zadnjim  $aR$ ?
    - $bR(0) \ aR \ aW \ aR \dots aW \ bW(1) \ aR(1) \text{ [vse ostalo] } aW(2)$**

# Letalski poleti

Podano je neuteženo drevo. V njem večkrat poišči neko vozlišče na razdalji  $d_i$  od vozlišča  $v_i$ .

- če poznamo najbolj oddaljeno vozlišče, samo sledimo poti
  - če je bližje kot  $d_i$ , ni rešitve
- A – B premer drevesa (najdaljša pot)
  - poiščemo ju lahko z dvema iskanjema v širino
  - najbolj oddaljeno vozlišče (X) bo vedno eno od krajišč premera





# Letalski poleti

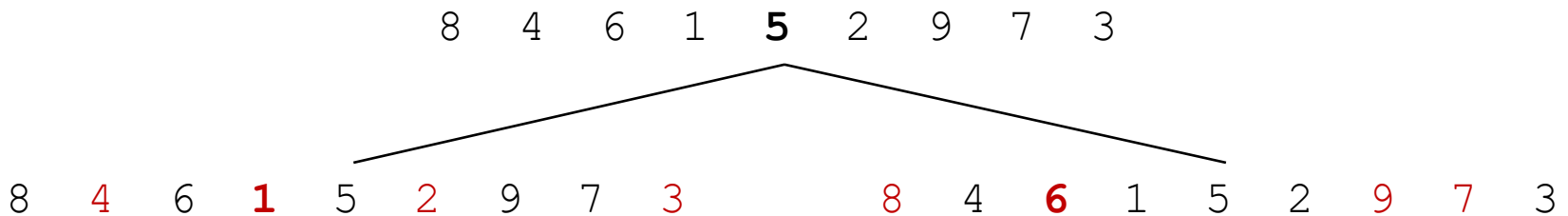
---

- najbolj oddaljeno vozlišče bo vedno eno od krajišč premera
  - dva ločena problema (koren v A in v B)
- iščemo  $d_i$ -to vozlišče na poti proti korenu
  - podobno kot pri problem LCA izračunamo  $2^k$ -te prednike
- izračun prednikov:  $O(n \log n)$
- odgovor na poizvedbo:
  - vedno krajši skoki proti korenu
  - $O(\log n)$

# Quicksort

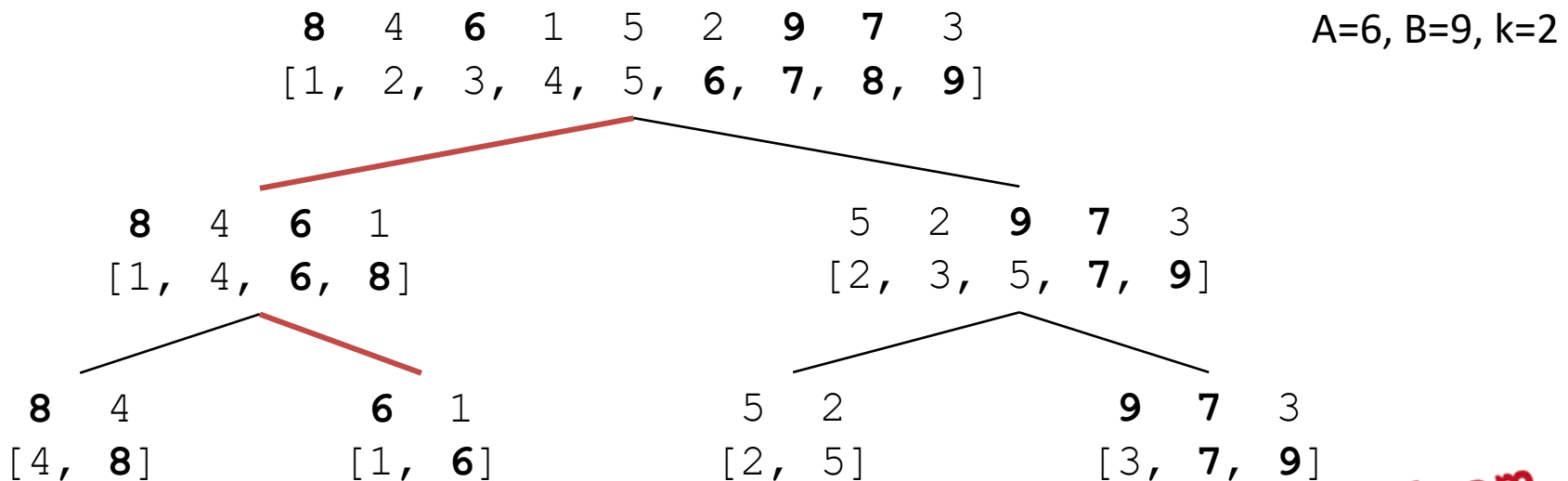
Izračunaj število operacij, ki jih potrebuje quicksort za urejanje seznama števil.

- naivna simulacija je prepočasna
  - quicksort v najslabšem primeru naredi  $O(n^2)$  operacij
- vedno urejamo nek interval vrednosti  $[A,B]$ , ki so razpršene po zaporedju
  - iščemo k-tega med vrednostmi  $[A,B]$ , ki predstavlja pivot



# Quicksort

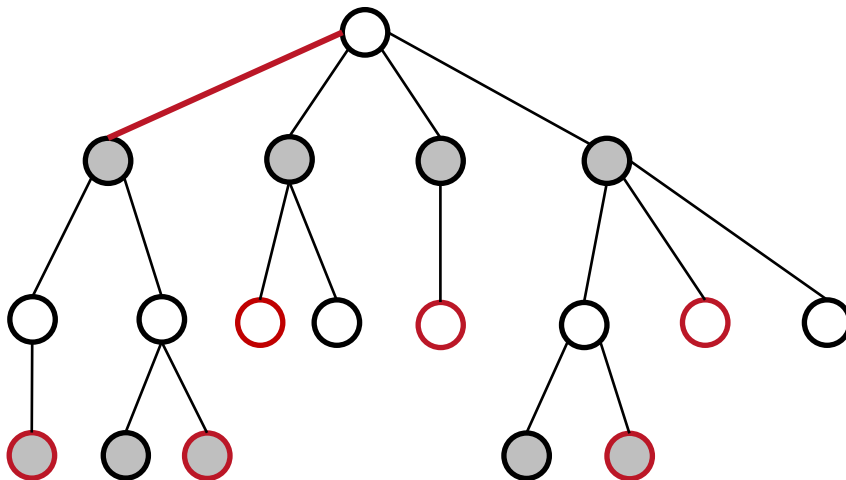
- segment tree: hranimo urejene sezname števil v območjih
- pozicija k-tega med števili na območju  $[l,r]$  z vrednostjo v  $[A,B]$ 
  - če jih je v levem otroku dovolj nadaljujemo tam, sicer na desni strani
    - z bisekcijo preštejmo količino števil do B in do A (razlika)



# Mat

Poišči mat v N potezah.

- ni kmetov (jemanje v stran, en passant, promocija)
- enostavno v teoriji, malo manj v praksi
  - min-max igra na drevesu stanj
  - komplikacija s pat-om



# Mat

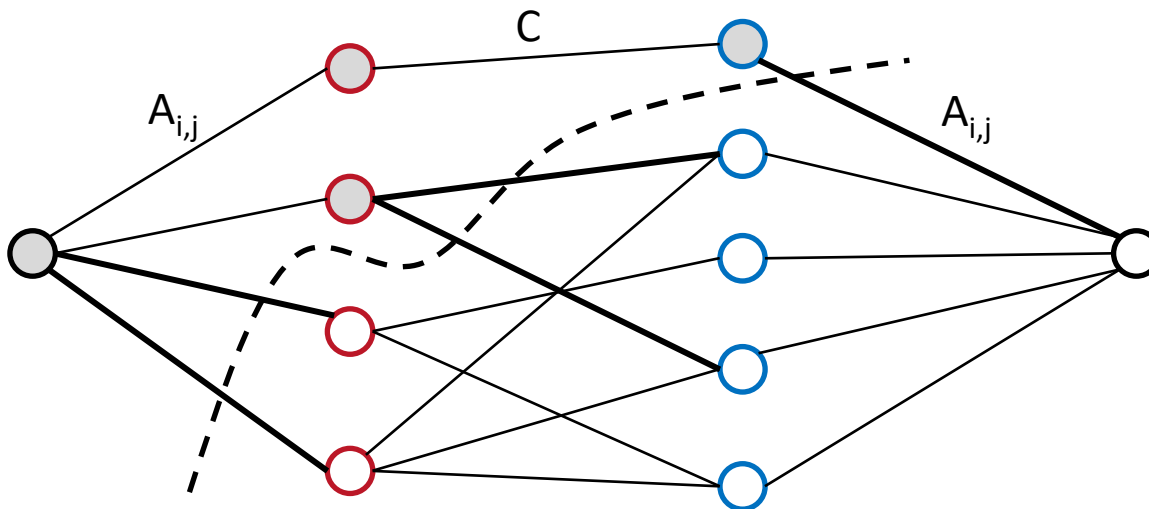
---

- sistematično reševanje (ni časa za refaktorizacijo in debugiranje)
  - stanje opišemo s seznamom figur (barva, tip, pozicija)
  - **play** (play\_white, play\_black) ... poteza igralca
    - **get\_states** ... seznam naslednjih stanj
      - **candidate\_moves** ... mogoči premiki figur iz (x1,y1) na (x2,y2)
        - **jump\_moves** ... premiki kralja, skakača
        - **line\_moves** ... premiki topa, lovca, dame (do vključno prve ovire, če je nasprotne barve)
      - **move** ... izvede potezo (premik figure in pobiranje)
      - **safe** ... je kralj varen v novi poziciji
        - preverimo *candidate\_moves* nasprotnika (lahko pristane kdo na kralju?)
    - če ni potez, je trenutno stanje varno (*safe*) ... mat, pat

# Košnja

Izravnaj travnik za čim hitrejšo košnjo.

- minimalni prerez (min cut) v “dvodelnem” grafu
  - nizke celice in visoke celice
  - povezava nizkih z izvorom, visokih s ponorom (cena spremembe višine  $A_{i,j}$ ) in sosednjih med seboj (cena prehoda s kosilnico  $C$ )
  - redek graf; Ford-Fulkerson, Edmonds-Karp, ...



# Zaključek

---

- hiter začetek
- precenili ste zahtevnost naloge Varnost na letališču
  - če se vam zdi kaj dvoumno, vprašajte za pojasnilo
- Predmetni moduli je konceptualno enaka kot Vsote števk s prejšnjega kola
- pasti:
  - Quicksort (MLE, TLE)
  - Vzporedno izvajanje (WA)
- pretoki, prerezi