CERC 2013

Rešitve nalog

Tomaž Hočevar[1] Nino Bašić[2]

[1]Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

[2]Fakulteta za matematiko in fiziko, Univerza v Ljubljani

5. december 2013

Darts (practice session)

Naloga

Podan imamo seznam zadetkov (koordinat) v tarčo, izračunati pa moramo število doseženih točk. Tarča je razdeljena na območja s koncentričnimi krogi polmerov $20, 40, \ldots, 200$, ki so vredni $200, 180, \ldots, 20$ točk.

Poiščeš najmanjši krog znotraj katerega je strel in prišteješ točke.

- preiskovanje celotnega seznama krogov
- bisekcija
- izračunamo krog v $O(1)$

camelCase (practice session)

Naloga

Poznamo tri oblike pisanja imen spremenljivk, ki so sestavljene iz večih besed: `PascalCase`, `camelCase` in `underscore_case`. Podan imamo niz v eni izmed teh oblik in ga moramo pretvoriti v ciljno obliko.

- enostavna obdelava nizov
- poljubna oblika → seznam besed
- seznam besed → poljubna oblika

Strips (practice session)

Naloga

Imamo mrežo velikosti $N \times M$, kjer eno polje manjka. Pokrili bi jo radi s ploščicami velikosti 3×1 oz. 1×3.
Zgled: $N = M = 5$, manjka $(3, 3)$

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \# & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 & 3 & 4 \\ 2 & 2 & 2 & 3 & 4 \\ 5 & 6 & \# & 3 & 4 \\ 5 & 6 & 7 & 7 & 7 \\ 5 & 6 & 8 & 8 & 8 \end{bmatrix}$$

- očiten pogoj: $3 \mid (MN - 1)$

$$\begin{bmatrix} \cdot & \cdot & \# & \cdot \end{bmatrix} \qquad \text{(ta pogoj ni dovolj za obstoj rešitve!)}$$

Strips (practice session) cont'd

- *diagonalna barvanja* s 3 barvami:

$$\begin{bmatrix} A & B & C & A \\ C & A & B & C \\ B & C & \# & B \\ A & B & C & A \end{bmatrix} \qquad \#A = \#B = \#C = 5$$

  Kakorkoli bomo že postavili ploščico na diagonalno pobarvano polje, bomo vedno pokrili vsako barvo po enkrat. Če ne velja $\#A = \#B = \#C$, potem rešitev ne obstaja.

- druga možnost (za barvanje iste mreže):

$$\begin{bmatrix} A & B & C & A \\ B & C & A & B \\ C & A & \# & C \\ A & B & C & A \end{bmatrix} \qquad \#A = 6, \#B = 4, \#C = 5$$

Strips (practice session) cont'd

Algoritem (greedy):

- če lahko odrežeš trak širine/višine 3, to tudi storiš
- ostanejo primeri največ 5×5, za katere na roke (oz. z "grobo silo") preveriš, kateri so rešljivi

Ali to deluje? Dokaz?

- za vsako mrežo $N \times M$ poiščemo možna mesta za manjkajoče polje:
  - diagonalno barvanje $\leftarrow$ s tem izločimo nekatere nemogoče možnosti
  - simetrije (rotacije in zrcaljenja plošče) $\leftarrow$ zmanjšamo si količino dela
- vse možnosti do 100×100, ki jih nismo izločili, poskusimo rešiti z zgornjim postopkom; na srečo se izkaže, da algoritem vedno najde rešitev

Alternativna možnost: 2-SAT (prepišeš iz zapiskov ;-))

Bus (72/81)

Naloga

Na vsaki postaji iz avtobusa z n potniki izstopi točno polovica potnikov in še pol potnika. Po k postajah je avtobus prazen. Koliko potnikov je bilo na začetku na avtobusu, če ni bil med izstopanjem nihče poškodovan?

- uganka
- $n' = n - (\frac{n}{2} + \frac{1}{2})$
- obrneš proces: $n = 2n' + 1$
- rezultat je $2^k - 1$ (indukcija)

What does the fox say? (71/130)

Odgovor na vprašanje se skriva tukaj :-)

Naloga

Imamo podan niz (posnetek oglašanja živali v gozdu):

`toot woof wa ow ow ow ...`

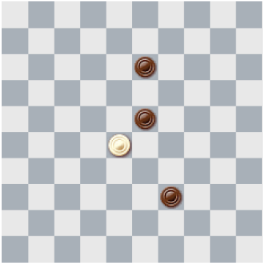
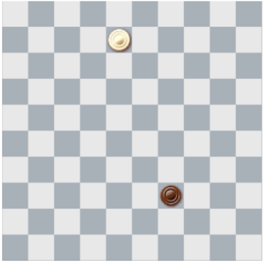
Podano imamo še, kako se oglašajo druge živali, v naslednjem formatu:

```
dog goes woof
fish goes blub
...
```

- trivialno – naredi, kar piše v nalogi
- seznam "tujih" besed
- besedo iz besedila izpišemo, če je ni na seznamu

Draughts (66/151)

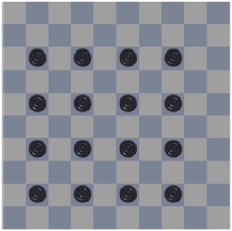
Naloga

- brute-force
  - bo dovolj hitro?
  - taktike: (a) poskusiš pa kar bo bo, (b) počakaš na druge ekipe, (c) izračunaš . . .

Draughts (66/151) cont'd

- najhujši scenarij:

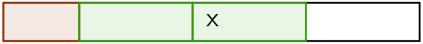


- $3^{16} \approx 43 \cdot 10^6$ $\leftarrow$ groba ocena; v praksi precej manj!
- rekurzija
- tabela možnih premikov (ne kopirat kode 4-krat)

Crane (48/71)

Naloga

Urediti želimo permutacijo števil $\{1, 2, ..., n\}$. V vsaki potezi izberemo interval sode dolžine ter zamenjamo prvo in drugo polovico intervala. Vrstni red števil znotraj polovic intervala se ohrani. Na voljo imamo $\approx 500\,000$ potez za $n \leq 10\,000$.



- seznam urejamo od leve proti desni
- število x v 1 koraku premaknemo v prvo polovico neurejenga dela
    - pazi na liho in sodo dolžino neurejenga dela
- v naslednjem koraku ga premaknemo na začetek
- časovna zahtevnost: $O(n^2)$, št. operacij $< 2n$

Digraphs (32/135)

Naloga

Podan imamo seznam prepovedanih parov črk (angleške abecede), npr.

```
aa az ba bb bc ...
```

Radi bi poiskali največjo kvadratno matriko (ki vsebuje črke), tako da se v nobeni vrstici in stolpcu (beremo od leve proti desni oz. od zgoraj navzdol) ne pojavijo prepovedani pari.

Zgled?

Digraphs (32/135) cont'd

Opažanja:

- če poznamo zunanji rob, lahko napolnimo notranjost (kjučna opazka)
- naloga se poenostavi na linearen problem
- pazi na primer, kjer je rešitev velikosti 1×1

Dve možnosti:

1. najdaljša pot v usmerjenem grafu:
   vozlišča = črke, povezave so med vsemi ne-prevedanimi pari črk
   - obstaja cikel?
   - DAG: topološko urejanje, najdaljša pot
   - $O(\Sigma^2)$
2. dinamično programiranje
   - $f(n, c)$ - obstaja pot dolžine n, ki se začne z znakom c
   - $O(n \cdot \Sigma^2)$ - počasneje ampak enostavnejša implementacija

Magical GCD (29/126)

Naloga

$MGCD(x_1, ..., x_k) = k \cdot GCD(x_1, ..., x_k)$ Podano je zaporedje števil $a_1, ..., a_n$. Poišči maksimalen MGCD njegovega strnjenega podzaporedja. Omejitve: $n \leq 10^5$, $a_i \leq 10^{12}$.

- Evklidov algoritem
  - $gcd(x, y) = gcd(y, x\%y)$
  - $gcd(x, y, z) = gcd(gcd(x, y), z)$
  - $O(\log x)$
- testiramo vsa podzaporedja?
- faktoriziramo števila a_i?
- fiksiraš konec in iščeš optimalen začetek intervala? ✓

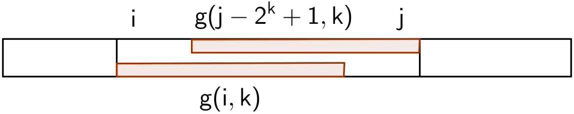
Pometanje

1	4	9	i = 11	
1	6	12	48	...

- naj bo konec podzaporedja na mestu i
- obstaja samo $O(\log a)$ vrednosti $gcd(a_x, ..., a_i)$ za $x \leq i$
    - gcd izgublja praštevilske faktorje
- $opt[g] = min(x \; ; \; gcd(a_x, ..., a_i) = g)$
- kako popraviš *opt* pri premiku konca iz i na $i + 1$?
    - zgradiš novega in združiš tiste z istim gcd-jem
    - map, seznam
- $O(n \log^2 a)$

Preskakovanje

- pripraviš teren za izračun gcd-jev vseh možnih podzaporedij v $O(\log a)$
  - $g(i, k) = gcd(a_i, ..., a_{i+2^k-1})$
  - $g(i, k) = gcd(g(i, k-1), g(i+2^k, k-1))$
  - čas, prostor: $O(n \log n)$
  - $gcd(a_i, ..., a_j) = gcd(g(i, k), g(j-2^k+1, k))$



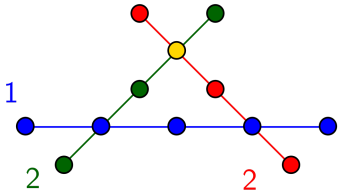
- naj bo konec podzaporedja na mestu i
- več zaporednih začetkov x bo dalo isto vrednost $gcd(a_x, ..., a_i)$
- enakovredne začetke preskočimo z bisekcijo
- $O(n \log n \log^2 a)$

Subway (12/48)

Naloga

Na podzemni železnici z n postajališči in l linijami, ki tvorijo skupaj e povezav, iščemo optimalno pot od postajališča x do y. Optimalna pot vsebuje *minimalno* število linij oz. prestopanj, med enakovrednimi pa iščemo tisto z *maksimalnim* številom prevoženih povezav. Omejitve: $n \leq 300\,000$, $l \leq 100\,000$, $e \leq 1\,000\,000$.

- oblika vhodnih podatkov
  - `split(str, ", ")` ali `str+=","`
  - postajališča prevedemo na števila $1..n$.
- podatkovne strukture
  - linija $\rightarrow$ postajališča, postajališče $\rightarrow$ linije
- iskanje v širino (BFS)
  - po linijah, ne po postajališčih
  - minimalno število prestopanj



Maksimalno število prevoženih povezav

- iščemo "najdaljšo" in ne najkrajšo pot
- precej načinov za WA
- rešujemo po linijah
- 3 tipi točk: vstopne, še neizračunane, delno izračunane z drugih linij
- iščemo max. razdaljo do neke vstopne točke
  - pometanje v levo in desno
  - ločeno izračunamo rešitev v vsako smer

Chain & Co. (11/25)

Naloga

Podanih imamo n ($n \leq 10^6$) enotskih kvadratkov v $\mathbb{R}^3$. Podani s z dvema nasprotnima ogliščema: (x_i, y_i, z_i) in (x_i', y_i', z_i'), kjer so koordinate cela števila in velja $-10^9 \leq x_i, y_i, z_i, x_i', y_i', z_i' \leq 10^9$. Vsaka stranica je vzporedna kateri od koordinatnih osi. Nobena dva kvadratka se ne sekata. Ali jih lahko razdelimo v dve množici A in B tako da velja: vsak $a \in A$ je neločljivo povezan z vsemi iz B in vsak $b \in B$ je neločljivo povezan z vsemi iz A?

- opis z grafom (kako velik je ta graf?)
- 3 tipi kvadratov: XY, YZ, XZ
- vzemimo $A = XY$, $B = YZ$; kaj more veljati, da bo pogoj naloge izpolnjen? razmislimo na tablo ...
- pazi na primere, ko je katere od množic XY, YZ, XZ prazna
- implementacija? (gre v 50 vrsticah)

Rubik's Rectangle (5/27)

Naloga

1	2	15	4
8	7	11	5
12	6	10	9
13	14	3	16

1	2	15	4
8	7	11	5
9	10	6	12
13	14	3	16

Operaciji: (a) flip vrstice, (b) flip stolpca
Omejitve: $1 \leq W, H \leq 100$

- delovanja permutacijskih grup :-)
- vsako št. lahko z danimi operacijami spravimo 4 lokacije (orbita) – če matrika ne vsebujejo pravih števil na pravih mestih očitno ni mogoče

Rubik's Rectangle (5/27) cont'd

Naloga

1	2	15	4
8	7	11	5
12	6	10	9
13	14	3	16

1	2	15	4
8	7	11	5
9	10	6	12
13	14	3	16

- poljubne tri lahko zaciklamo $(R_i C_j R_i C_j)$ in dobimo s tem vse sode permutacije (alternirajoča podgrupa)
- če obrnemo katerikolo vrstico ali stolpec (še ena dodatna transpozicija), spremenimo parnost pripadajočim četvorkam

- izmed prvih h/2 vrstic in w/2 stolpcev bi radi obrnili nekatere, da bodo vse četvorke sode
  - sistem enačb modulo 2 – Gaussova eliminacija
  - odločitev o izbiri prve vrstici definira vse ostalo (predpostavimo, da ni izbrana, ker lahko izbor invertiramo in bo ok)
- implementacija
  - najkrajše poti (npr. kako pridem do permutacije 1, 3, 2, 4 pri čemer so obrnjeni spodnja vrstica in oba stolpca)
  - pazi na sredinske stolpce/vrstice pri lihih dimenzijah

Captain Obvious and the Rabbit-Man (1/16)

Naloga

Naj bodo F_i Fibonaccijeva stevila ($F_1 = 1$, $F_2 = 2$, $F_i = F_{i-1} + F_{i-2}$). Nekdo si je izmislil števila $a_1, a_2, \ldots, a_k$, ki jih ne poznamo. ($k \leq 4000$, $M \leq 10^9$).

$$p(i) = a_1 \cdot F_1^i + a_2 \cdot F_2^i + \ldots + a_k \cdot F_k^i \pmod{M}$$

Podane imamo vrednosti $p(1), \ldots, p(k)$. Iščemo $p(k+1)$.

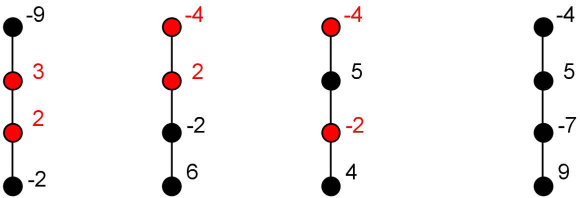
- sistem enačb modulo neko praštevilo m
- Gaussova eliminacija
  - modularni inverzi (x^{m-2})
  - $O(k^3)$ - prepočasi

Escape (0/21)

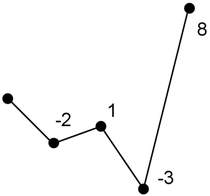
Naloga

Podano je drevo z $n \leq 200\,000$ vozlišči. Pri prvem vstopu v vozlišče i se naša "moč" poveča oz. zmanjša za h_i. Ali lahko pridemo od vozlišča 1 do nekega ciljnega vozlišča t, ne da bi pri tem naša moč padla pod 0?

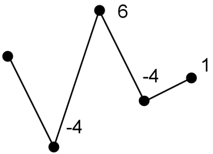
- namesto dosegljivosti vozlišča t iščemo max. moč (t-ju pripnemo vozlišče $+\infty$)
- problem je enostaven, če je drevo pot
- "normalna" pot
  - alternirajoča negativna in pozitivna vozlišča
  - pozitivno vozlišče je "večje" od predhodnjega negativnega
  - negativna vozlišča "naraščajo"



Združevanje dveh parov vozlišč: $(-a, b, -c, d) \rightarrow (-x, y)$



$-4, 8$



$-4, 3$

- drevo lahko prevedemo na “normalno” pot
  - rekurzivno prevedemo otroke
  - poti otrok bomo obiskovali po vedno bolj negativnih vozliščih (rezultat bo “normalna” pot)
  - zaradi korena je treba normalizirati samo prvih nekaj parov v poti
- implementacija
  - delamo lahko s pari točk: (negativna, pozitivna)
    - pred pozitivno točko lahko vstaviš -0
    - za negativno točko lahko vstaviš $+0$
  - vrstni red parov v “normalni” poti je impliciten (multiset)
  - združevanje poti: pare iz krajše vstavljamo v multiset daljše poti
  - pri dodajanju korena popravljamo najmanjše pare v multiset-u
- časovna zahtevnost: $O(n \log^2 n)$

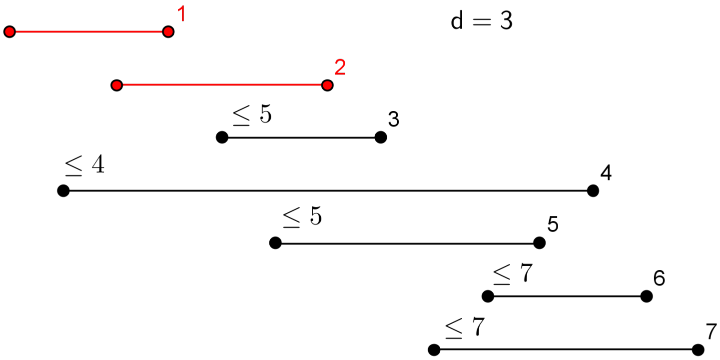
History course (0/4)

Naloga

Množico daljic na premici bi radi zložili v nek optimalen vrstni red. Če se daljici ne sekata, morata biti v naravnem vrstnem redu. Sicer pa iščemo minimalen d, da bo v optimalnem vrstnem redu razdalja med vsemi pari sekajočih se daljic $\leq d$.

- kompresija koordinat
- bisekcija po razdalji d
  - vsaka daljica ima omejitev pozicije, do katere se mora pojaviti
  - najdemo kritični nivo (najmanjšo pozicijo, kjer daljice povsem zapolnijo prosta mesta)
  - optimalno je izbrati daljico z najbližjim koncem
    - vpliva na najmanj drugih

Primer rešitve



- kritični nivo: ≤ 5
- optimalna kritična daljica: 3

- pazi, odgovor je lahko 0
- časovna zahtevnost: $O(n \log^2 n)$
  - $O(n^2 \log n)$ - prepočasi
  - drevesni strukturi za iskanje kritičnega nivoja in optimalne kritične daljice

Implementacija

- range tree, ki nam za interval nivojev vrne optimalno daljico
  - za vsak nivo hranimo pripadajoče daljice urejene po koncih
  - nad tem zgradimo range tree za poizvedbe v intervalih
- range tree za iskanje kritičnega nivoja
  - i-ti element predstavlja št. prostih mest od 1 do i (vrednost $0 \rightarrow$ kritični nivo)
  - sprememba nivoja daljice iz a na b: odštej 1 vsem na tem intervalu
  - lazy update
    - hranimo najmanjšo pozicijo ničle in najmanjšo neničelno vrednost