Univerzitetni programerski maraton

FINALE 2023 – rešitve nalog

Tomaž Hočevar

</upm UNIVERZITETNI PROGRAMERSKI MARATON

Zdravilo

Preštej pojavitve vzorca T v besedilu S s K dovoljenimi napakami.

- Primer: S= CGCCCGAATCCAGAACGCATT T=ATTCTGGA K=3

  CGCCCG**AATCCAGA**ACGCATT

  CGCCCGA**ATCCAGAA**CGCATT

- preveri vseh |S|-|T|+1 možnih poravnav
- izračunaj število neujemajočih se znakov
- preštej sprejemljive poravnave
- O(nm)

</upm UNIVERZITETNI PROGRAMERSKI MARATON

URL naslovi

Iz naslova članka sestavi del URL naslova z zamenjavo posebnih znakov z vezaji in dodjanjem št. pripon.

- primer
  - `Studentje si zelijo prihranek casa, nizko ceno in okusen obrok, temu pa se prilagaja tudi ponudba`
  - `studentje-si-zelijo-prihranek-casa-nizko-ceno-in-okusen-obrok-temu-pa-se-prilagaja-tudi-ponudba`
- povečujemo pripono
- pozor:
  - več zaporednih posebnih znakov
  - naslov lahko že obstaja (s pripono ali brez)

Skrito zaporedje

Določi nadaljevanje skritega zaporedja.

- iteraktivna naloga
- zaporedje (n-ti člen) je oblike $an^2 + bn + c$ ali rq^n za neke konstante a, b, n, c, q, r ϵ [-20, 20]
- preveri vse možne nabore konstant
  - ali ustrezajo dosedanjim členom?
  - kakšen bi bil naslednji člen?
- če je naslednji člen enoličen, zaključi

- primeru 6, 11, 18 ustreza samo oblika a=1, b=2, c=3, ki mu sledi število 27

</upm UNIVERZITETNI PROGRAMERSKI MARATON

Dendrogram

Izriši podano drevo v obliki dendrogama.

- obdelaj opis strukture
  - oklepaju lahko poiščemo pripadajoč zaklepaj
- izriši strukturo v obliki dendrograma
  - pravila so jasno opisana
  - izris naj vrača vrh dendrograma

```
(((X,(Y,Z)),((A,B),C)),D)
```

```
X---|
    |-|
Y-| | |
  |-| |
Z-|   |-|
        | |
A-|     | |
  |-| | | |
B-| |-| | |
    |   |
C---|   |
        |
D-------|
```

</upm UNIVERZITETNI PROGRAMERSKI MARATON

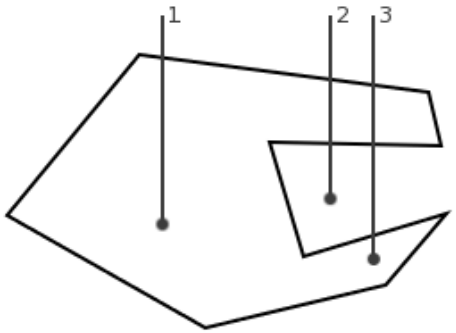
Pokrajine

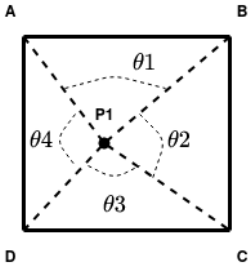
Določi lokacijo točke v particiji ravnine na večkotnike.

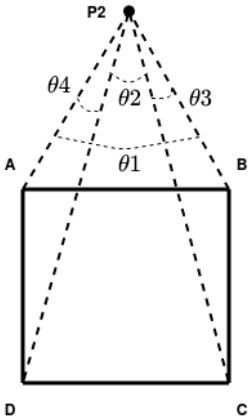
- point-location problem
  - naključne lokacije gob (brez posebnih primerov)
- za vsako točko lahko preverimo vseh 12 regij
- ali je točka v večkotniku (point in polygon)?

a) ray casting

b) winding number (angle sum)







</upm UNIVERZITETNI PROGRAMERSKI MARATON

Parsanje stringov

Ugotovi, ali niz ustreza “Python” sintaksi.

- delna Python sintaksa, ~~eval()~~
  - “abc” “def”, “\u0391”, ...
- najdi začetne narekovaje
- pregleduj preostanek niza po znakih in računaj dolžino
  - končni narekovaji: je še kaj ostalo?
  - ubežna koda \
    - naslednji je znan: \’”abfnrt
    - naslednji je x, mu sledita dva šestnajstiška?
    - naslednji je osmiški znak, morda celo dva ali trije?
    - naslednji je neprepoznan
  - normalen znak

</upm UNIVERZITETNI PROGRAMERSKI MARATON

Hiperpot do šole

Simuliraj pot po hipermreži v skladu s semaforji.

- v vsaki dimenziji je definirana smer (proti šoli – Manhattan)
- katera dimenzija ima prva zeleno luč?
  - t % (r+z) < r ---> rdeča
  - sicer ---> zelena

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

  - n·d korakov, vsakič preverjanje d dimenzij … TLE

- perioda semaforja <=10, perioda mesta LCM(1,2,…,10) = 2520
  - urejen preferenčni seznam smeri za vsak čas t <= 2520
  - (čakalni čas, smer)

</upm UNIVERZITETNI PROGRAMERSKI MARATON

Čarobni napis

V nizu S zamenjaj vprašaje s črkami, da bo čim več pojavitev vzorca P.

AB?B**AB?A**A?BBA??A
ABBAAC

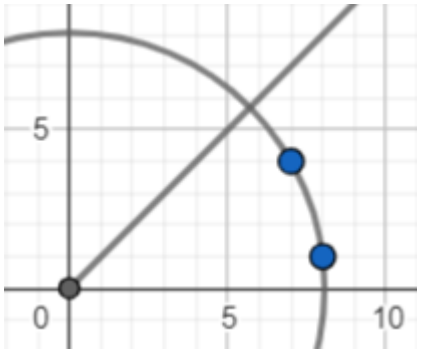
- recimo, da menjamo vprašaje po vrsti
- dinamično programiranje
  - f(i, rob) = max. št. pojavitev, ki vključujejo tudi znake od i-tega naprej, če se S[:i] konča z nizom rob.
  - rob = najdaljši prefix vzorca P, ki je suffix niza S[:i] oz. njegova dolžina
- "KMP", $O(n^2)$
- kaj naredimo glede na znak S[i]?
  - če je vprašaj, preverimo vse možnosti
  - kako se spremeni rob?
    - vnaprej izračunamo spremembe: g(rob, c)

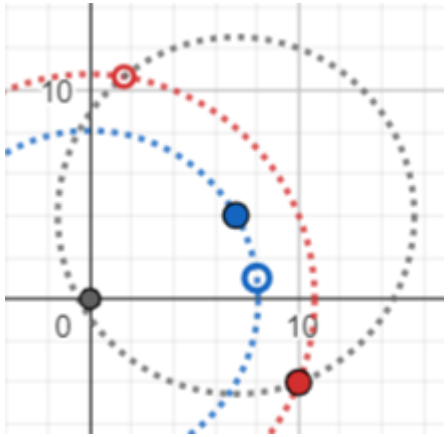
</upm UNIVERZITETNI PROGRAMERSKI MARATON

Zastave

Razpostavi zastave na celoštevilske koordinate v skladu s podanimi medsebojnimi razdaljami.

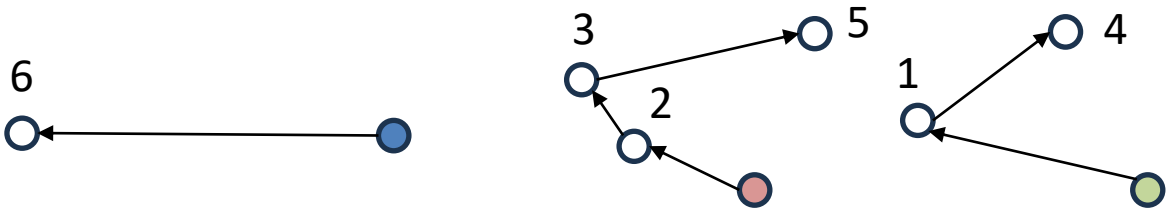
- prvo zastavo postavimo v izhodišče
- preverimo vse možne lokacije druge zastave
  - seznam odmikov (dx_i, dy_i) glede na razdaljo i-te zastave do prve
  - en od odmikov ne bo presegal korena razdalje
- postavimo tretjo zastavo (in naslednje)
  - gledamo možne odmike glede na prvo
  - preverjamo razdaljo do druge
  - obstajata kvečjemu dve možnosti
- pazimo na zastave na razdalji 0





Nujna pomoč

Optimalno razporedi ekipe nujne pomoci na koordinate, kjer potrebujejo pomoč, da bo prepotovana pot čim krajša.



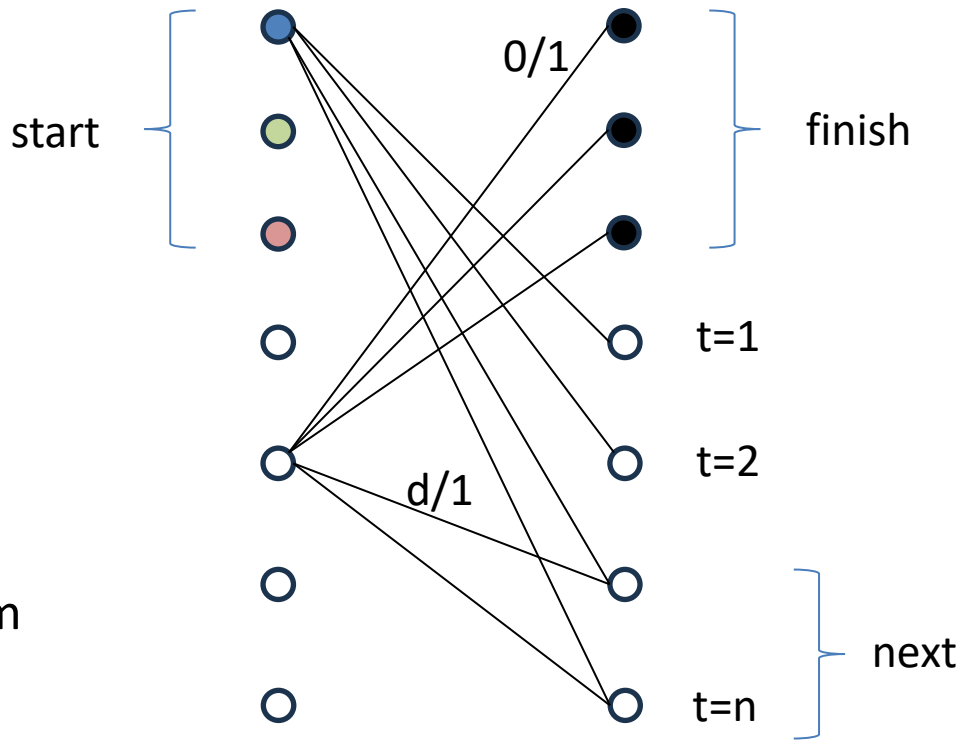
- požrešna strategija ni ok
- dinamično programiranje?
  - stanje = (čas, lokacije vseh ekip) … O(n^k)
  - prepočasi
- vsaki lokaciji moramo prirediti naslednjo

</upm
UNIVERZITETNI
PROGRAMERSKI MARATON

Nujna pomoč

- assignment problem
  - Hungarian algorithm
  - min cost max flow

- min cost (max) flow
  - source, sink
  - successive shortest path
    - "Ford Fulkerson"
    - najkrajša pot s pozitivnim pretokom
    - pazi: negativne uteži
      - Bellman-Ford
    - $O(n^4)$, v praksi hitrejši



</upm
UNIVERZITETNI PROGRAMERSKI MARATON