

Algoritmi in podatkovne strukture 1

15. februar 2024

Ime in priimek: _____

Vpisna številka: _____

Pri reševanju si lahko pomagate z enim A4 listom zapiskov.

Svoje odgovore pišite jasno in jih utemeljite.

Ocenjuje se odgovore na izpitni poli. Dodatni listi so namenjeni pomoči pri reševanju in jih ne oddate.

Čas reševanja: 90 min.

| | | | | | |
|---------|----|----|----|----|--------|
| Naloga: | 1 | 2 | 3 | 4 | Skupaj |
| Točke: | 25 | 25 | 25 | 25 | 100 |
| Ocena: | | | | | |

1. Za pripravo na izpit iz APS1 boste uporabljali kviz, kjer rešujete naloge iz n različnih tem. Teme si sledijo po vrsti, kot ste jih obravnavali na predavanjih. Da lahko rešite nalogo, ki se nanaša na i -to temo, morate pred tem rešiti vsaj eno nalogo iz vsake teme od 1 do $i - 1$. Ni pa vam treba vsakič izbrati nove teme. Če ste že rešili vse teme do vključno teme i , lahko svoje znanje utrjujete tudi z reševanjem naloge iz katere od že rešenih tem $j \leq i$. Število nalog pri vsaki temi je zelo veliko. Pri rešitvi prve naloge iz i -te teme svoje znanje povečate za $a_i > 0$ točk. Ob vsaki nadaljnji rešitvi naloge za utrjevanje snovi i -te teme pa za $b_i > 0$ točk. Do izpita imate časa za rešitev k nalog. Za največ koliko točk lahko v tem času povečate svoje znanje?

[10] (a) Opišite čim bolj učinkovit algoritem, ki izračuna odgovor na zastavljeno vprašanje. Navedite in utemeljite tudi njegovo časovno in prostorsko zahtevnost.

[15] (b) V C++ implementirajte funkcijo `kviz`, ki bo sprejela število rešitev k in seznam parov (a_i, b_i) , ki predstavljajo število točk za rešitev naloge iz i -te teme. Vrne naj iskano največje število točk.

Implementacija lahko ustreza manj učinkovitemu postopku od prej opisanega, vendar bo zato prejela manj točk (v tem primeru jo tudi na kratko opišite). Pri implementaciji lahko uporabljate vso funkcionalnost standardne knjižnice C++.

```
1 vector<int> tocke = {{2,1}, {2,9}, {8,6}, {7,10}, {7,8}, {9,8}, {8,6}};
2 int x = kviz(6, tocke);
3 // 40
4 // resimo 1. temo, resimo 2. temo in se 4x utrjujemo znanje na 2. temi
```

2. Podan je bil urejen seznam n števil x_1, x_2, \dots, x_n . Nekdo je na tem seznamu zaporedoma izvedel manjše število k ($k \ll n$) operacij, kjer je v vsaki operaciji izbral indeks i in med seboj zamenjal sosednji števili na indeksih i in $i + 1$. Tako je v eni operaciji iz seznama

$$x_1, \dots, x_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, x_{i+2}, \dots, x_n$$

nastal nov seznam

$$x_1, \dots, x_{i-1}, \mathbf{x}_{i+1}, \mathbf{x}_i, x_{i+2}, \dots, x_n.$$

Kako se na takih skoraj urejenih seznamih, ki so rezultat k zamenjav sosednjih elementov, v najslabšem primeru obnesejo spodnji algoritmi za urejanje? Pri tem indeksov zamenjav i ne poznamo. Za vsak algoritem napišite in utemeljite njegovo časovno zahtevnost v odvisnosti od parametrov n in k .

- [5] (a) urejanje z izbiranjem (*selection sort*)
- [5] (b) urejanje z vstavljanjem (*insertion sort*)
- [5] (c) urejanje z zamenjavami (*bubble sort*) z zgodnjo ustavitvijo, ko ni več zamenjav
- [4] (d) hitro urejanje (*quick sort*) z izbiro prvega elementa za pivot
- [3] (e) urejanje z zlivanjem (*merge sort*)
- [3] (f) urejanje s kopico (*heap sort*)

3. Odgovorite na vprašanja o Kruskalovem algoritmu. Svoje odgovore utemeljite.

- [3] (a) Na kratko predstavite problem, ki ga rešuje.
- [4] (b) Na kratko predstavite Kruskalov algoritem.
- [4] (c) Kakšna je časovna zahtevnost algoritma na polnih grafih z n vozlišči?
- [4] (d) Kakšna je časovna zahtevnost algoritma na redkih grafih z n vozlišči in e povezavami?
- [10] (e) Kako bi prilagodili algoritem za uporabo na grafih z majhnimi celoštevilskimi utežmi povezav, npr. z intervala $[1, k]$? Kakšna bi bila njegova časovna zahtevnost?

4. Podan je bil povezan usmerjen utežen graf G z n vozlišči in samimi pozitivnimi utežmi. Nek zlobnež pa nam je graf zakompliciral z dodatkom več negativnih ciklov. Večkrat je izbral neko zaporedje vozlišč in iz njih ustvaril negativen cikel. Pri tem je dodal povezave med sosednjimi vozlišči tega zaporedja (ter iz zadnjega do prvega vozlišča) z različnimi utežmi. Nekatere uteži so morale biti očitno negativne, da je nastal negativen cikel. Na koncu postopka smo dobili graf z e povezavami. Na našo srečo pa je označil izbrana vozlišča. Za vsako vozlišče torej vemo, ali je del nekega negativnega cikla ali ne. Ne poznamo pa dejanskih negativnih ciklov, ki so bili dodatni v graf. Način predstavitve grafa in označenih vozlišč lahko izberete sami, vendar ne pozabite tega opisati. Rešite sledeči nalogi in utemeljite svoje odgovore ter navedite časovne in prostorske zahtevnosti.

- [12] (a) Predstavite čim bolj učinkovit algoritem, ki bo v takem grafu ugotovil, ali je najkrajša pot iz vozlišča A do B nedefinirana (neskončno majhna) ali ne.
- [13] (b) Predstavite čim bolj učinkovit algoritem, ki bo v takem grafu izračunal dolžino najkrajše poti, če veste, da je dobro definirana (ni neskončno majhna), kar bi lahko ugotovili z rešitvijo prejšnje naloge.