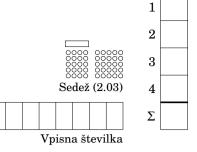
# Podatkovne strukture in algoritmi 1: Izpit 1

20. januar 2023

Čas reševanja je 120 minut. Pojasnite vse odgovore.



Ime in priimek

## 1. naloga (20 točk)

Na zabavo pride n oseb. Oseba je superzvezda, če jo poznajo vsi ostali na zabavi, medtem ko superzvezda ne pozna nobenega.

a) (3 točke) Pokaži, da na zabavi obstaja največ ena superzvezda.

Protislouje.

**b)** (17 točk) Na voljo imamo metodo pozna(a,b), ki vrne true, v kolikor oseba a pozna osebo b, in false sicer. Opiši algoritem, ki ugotovi, če na zabavi obstaja superzvezda. Za vse točke mora algoritem delovati v O(n)

O(NZ): preveriono za usakego, ali je superzvezda

Dve for zamki => O(NZ)

 $J(n\log)$ : Razdelino na pol in redurziono regimo.

Dobimo 2 keundidata (en levi, en desni)

in Se preverimo ali sta superzuezdi o  $\sigma(n)$ .

=) T(n) = 2T(n/2) + J(n)=)  $T(n) = O(n\log n)$ 

J(n): Damo use osebe na stelad S.

Dokler 1s1 ≥ 2 jemlemo iz stelada po

Jue osebi in preverjamo ali se poznata.

V usakem primen Ghko usu; emo osebo

Zavrzcmo kot super zuezdo.

Če na steladu ostano ena oseba, preverimo

J(n), če je res super zuezda.

#### 2. naloga (20 točk)

V nekem podjetju so zadolženi za testiranje posebnih čipov. V pomoč jim je posebno vezje, kamor priklopijo dva izmed čipov ter vsak čip pove za drugega, ali je dober ali slab. Dobri čipi vedno vrnejo pravilen odgovor, medtem ko slabim čipom ne moremo zaupati (lahko povejo bodisi pravilno bodisi narobe).

a) (2 točki) Vsi možno rezultati enega testa čipov A in B so zapisani v spodnji tabeli. Izpolni zadnji stolpec.

	$\operatorname{cip} A \text{ vrne}$	$\operatorname{cip} B \text{ vrne}$	Kaj lahko sklepamo o A 1	n <i>B</i> ?	
(O,O)	<i>B</i> je dober	A je dober	Oba dobra ali oba	slabci	
( D ,5 )		A je slab			
(5,0)	B je slab	A je dober	B slub	ζ	vedno Usuj en slab
(5)	B je slab	A je slab	usaj en slab	)	, J

**b**) (15 točk) Izmed n čipov za katere vemo, da je vsaj  $\frac{n}{2}$  dobrih želimo najti enega, ki je zagotovo dober. Opiši postopek, ki najde dober čip.

**Namig**: Z uporabno največ  $\lfloor \frac{n}{2} \rfloor$  prevedi zgornji problem na problem velikosti približno  $\frac{n}{2}$ . Za začetek lahko tudi predpostaviš  $n=2^k$  ter nato posplošiš na poljuben n.

c) (3 točke) Izmed n čipov za katere vemo, da je vsaj  $\frac{n}{2}$  dobrih poišči vse tiste čipe, ki so dobri v času  $\Theta(n)$ .

Po tocki b) imamo 
$$T(y) - T(y/2) + O(y)$$

=>  $T(y) - O(y)$ .

ko enterest imamo dober cip cubbo potestiramo vsc ostale u  $\Theta(y)$ .

Zakci, tega niste resevali?

## 3. naloga (20 točk)

Imamo n letališč oštevilčenih z  $\{0,1,\ldots,n-1\}$  ter metodo odhodni\_leti(letalisce, cas). Vsaka povezava je oblike (začetno\_letališče, čas\_odhoda, končno\_letališče, čas\_prihoda). Poleg tega so nekatera letališča večja ter zato potrebujemo več časa, da pridemo iz enega konca letališča na drugega. Za vsako letališče i imamo torej podan še najmanjši čas postanka na letališču i. Opiši algoritem, ki za podano začetno in končno letališče izračuna najkrajši čas potovanja od začetnega do končnega vozlišča. Kakšna je njegova časovna zahtevnost?

Malificiramo Djikstro, da upostevo se postenke

ua vozlistih/letalistih.

To najlazje storimo tako, da cenijčasu povezave

a >> b pristejemo postanekloj.

Pomembno: Grafa ne moramo zgraditi unaprij kor

ne vemo izhalnih povezav iz letalista

doklar tja ne pridemo v nekem časv til

kav nekaj vas je dela 6 to napoko, druguče ste

pa to nabogo reševali dobro.

#### 4. naloga (20 točk)

Opiši algoritem, ki izračuna najmanjšo dolžino particije niza S v palindromske podnize. Na primer:

- S = 'aabab' lahko razdelimo v 'aa|bab'. Ker S ni palindrom je torej najkrajša dolžina 2.
- S = 'ababbbabbababa' lahko razdelimo v 'a|babbbbab|b|ababa'. Dolžina je torej 4.

Bonus: Če algoritem deluje v 
$$O(n^2)$$
, dobiš dodatnih 8 točk.

 $O(N^3)$ :

 $N = (en(S))$ 
 $V = (e$ 

lmamo nº stani, in O(n) za izračun stanja => O(n3)

O (N2):

$$P[i][i] = true$$

$$P[i][j] = (S[i] == S[j]) AND p[i+1][j-1]$$

$$N^{2} stan_{j}, \sigma(1) \neq a stan_{j}e \Rightarrow \sigma(n^{2}),$$

Sedaj: dp[i] = najmanjša dolžinu particije niža S[ii]v palindromske podniže.

$$d_{\rho}[i] = \begin{cases} 1 & \text{if } \rho[o][i] = \text{true} \\ w_{i} & \text{if } \int_{\rho}[i] + 1 \\ j = 0, \dots, i-1 \end{cases}$$
and  $\rho[j+1][i] = \text{true}$