

Trajanje: 120 minuta. Rješenja zadataka na košuljici i po potrebi na zasebnim papirima. Obvezno zaokružiti na košuljici redni broj rješavanog zadatka (1–10).

21 1. (3 boda) Formalno definirati *Adwords* problem. Opisati *BALANCE* algoritam (navesti pseudokod).

21 2. (3 boda) Tri oglašivača A , B , i C natječu se za prikaz oglasa. Svi imaju isti budžet koji iznosi 3. Svaki oglašivač ponudio je 1 za prikaz oglasa. Oglašivači A i B natječu se za oglase tipa X i Y , dok se oglašivač C natječe za oglase tipa X . U sustav oglašavanja dolazi sljedeći niz korisničkih zahtjeva: $Y Y X X X Y Y Y$. Pretpostavite da sustav oglašavanja koristi *BALANCE* algoritam. Opisati rad algoritma nad navedenim nizom korisničkih zahtjeva. Odrediti kompetitivni omjer algoritma (engl. *competitive ratio*) za navedeni ulaz.

3. (3 boda) Odredite stanje struktura podataka nakon prvog i drugog koraka rada algoritma *PCY* (Park, Chen, Yu) za zadani skup košara. Pretpostaviti da se elementi u košarama indeksiraju abecednim redoslijedom od početnog indeksa 0. Funkcija sažimanja (engl. *hash function*) zadana je formulom: $(i + j) \% 4$, pri čemu su i i j indeksi elemenata u košari. Prag potpore (engl. *support threshold*) iznosi 4.

$$\begin{array}{ll} B_1 = \{a, b, c, d\} & B_4 = \{a, c, d\} \\ B_2 = \{a, c, d\} & B_5 = \{b, c, d\} \\ B_3 = \{a, b, d\} & B_6 = \{c, d\} \end{array}$$

4. (3 boda) Opišite Toivonenov algoritam. Definirajte pojam negativne granice (engl. *negative border*). Konstruirajte negativnu granicu za zadane košare uz prag potpore (engl. *support threshold*) $s = 3$.

$$\begin{array}{ll} B_1 = \{a, b, c\} & B_4 = \{a, b, c\} \\ B_2 = \{a, b\} & B_5 = \{a, c\} \\ B_3 = \{a\} & B_6 = \{b\} \end{array}$$

5. (3 boda) Za zadani graf na slici 1 napisati jednadžbe toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu u grafu (eng. *Flow Equation Formulation*). Analitičkim postupkom riješiti sustav jednadžbi. Napisati jednadžbu u matricnom obliku (eng. *Matrix Formulation*). Metodom uzastopnog potenciranja izračunati vrijednost rang vektora r za prve tri iteracije algoritma. Rang vektor u početku treba inicijalizirati tako da svi čvorovi dobiju jednak rang/utjecaj. Izračunati prosječnu apsolutnu kvadratnu pogrešku *RMSE* (eng. *Root Mean Square Error*) metode uzastopnog potenciranja kroz tri iteracije u odnosu na egzaktno analitičko rješenje.

6. (3 boda) Za zadani graf na slici 2 napisati jednadžbe toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu u grafu (eng. *Flow Equation Formulation*).

a.) Primjeni li se metoda uzastopnog potenciranja s ciljem izračunavanja vektora ranga r na zadani graf, hoće li dobiveni rezultati biti vjerodostojni? Detaljno objasniti odgovor.

b.) Modificirati zadani graf uvođenjem teleportirajućih poveznica (eng. *teleports*). Napisati vrijednost *Google* matrice A ako vjerojatnost da će slučajni šetač (eng. *random walker*) slijediti poveznicu iz grafa iznosi $\beta = 0.8$.

7. (3 boda) Analiziramo tok podataka koji se sastoji od e-mail adresa. Uniformne *hash* funkcije h_1, h_2, \dots, h_{10} preslikavaju adrese u 32-bitne brojeve. Funkciju h_1 koristimo za uzorkovanje: radi ubrzanja analize, u uzorak ulaze samo adrese x za koje je $h_1(x) < 10^8$. Funkcije h_2, h_3, \dots, h_9 koristimo za Bloomov filter u koji ubacujemo adrese uzorka. Funkciju h_{10} koristimo za Flajolet-Martinovu procjenu broja različitih elemenata, pri čemu je r = maksimalan broj nula na kraju nekog $h_{10}(x)$ iz uzorka.

a.) Procijenite broj različitih adresa u uzorku ako je $r = 25$.

b.) Procijenite broj različitih adresa u uzorku ako je popunjenost (gustoća) Bloomovog filtera 10%.

c.) Na temelju prethodnih dviju procjena p_a i p_b procijenite broj različitih adresa u cijelom toku podataka.

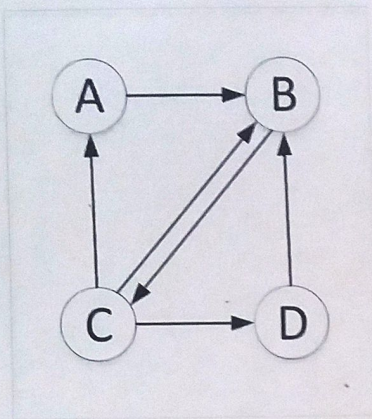
8. (3 boda) Senzor za mjerenje temperature zraka šalje odgovarajuća mjerenja s preciznošću od 0.1°C . Napišite (u pseudokodu ili jeziku po želji) algoritam koji za svaku pristiglu temperaturu odgovara na sljedeći upit: Koliko je puta u prethodnih 10^6 mjerenja izmjerena ista temperatura kao sada? Algoritam mora biti egzaktn (ne samo približno točan), smije koristiti poznate strukture podataka i mora biti vremenski i memorijski što efikasniji. Odredite vremensku složenost algoritma po pojedinom upitu.

9. (3 boda) Za zadani graf na slici 3 skicirati strukture podataka r^{t+1} , M i r^t koje se koriste za izračunavanje ranga (utjecaja) čvora u grafu pod pretpostavkom da:

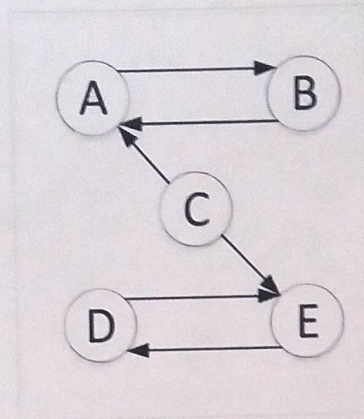
a.) Rang vektor r može stati u radnu memoriju, a matrica M ne može stati u radnu memoriju.

b.) Niti rang vektor r , niti matrica M ne mogu stati u radnu memoriju. U radnu memoriju može stati najviše $k = 2$ elemenata rang vektora. Skicirati prilagođeni zapis matrice M koji omogućuje manji broj iteracije kroz čitavu matricu prilikom računanja rang vektora.

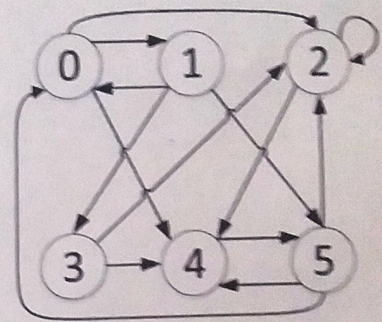
10. (3 boda) Pretpostavimo da se u nekom sustavu entiteti predstavljeni skupovima duljine 6 redaka. Od ukupno 6 redaka u skupu, 4 retka imaju vrijednost 0, a ostala dva retka imaju vrijednost 1. Nadalje, za generiranje sažetka skupa koristi se algoritam *MinHash*. Primjetite da ukupno postoji $6! = 720$ permutacija 6 redaka. Prilikom stvaranja sažetka odabire se indeks prvog reda u permutiranom poretku koji ima vrijednost 1.
- a.) Koliko će od ukupno 720 permutacija imati vrijednost MinHash sažetka imati 6?
- b.) Koliko će od ukupno 720 permutacija imati vrijednost MinHash sažetka imati 5?
- c.) Koliko će od ukupno 720 permutacija imati vrijednost MinHash sažetka imati 4?



Slika 1: Graf uz zadatak 5



Slika 2: Graf uz zadatak 6



Slika 3: Graf uz zadatak 9