Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Analiza velikih skupova podataka

Autori: Goran Delač, Marin Šilić i Klemo Vladimir
Ak. god. 2019/2020

4. Laboratorijska vježba

Treća laboratorijska vježba bavi se tematikom algoritama za obradu velikih grafova. U sklopu vježbe postoje dva zadatka. U prvom zadatku (zadatak A) potrebno je za ulazni usmjereni graf izračunati rang (eng. *Node rank*) svakog čvora u grafu. U drugom zadatku (zadatak B) potrebno za svaki čvor u neusmjerenom grafu izračunati određeno svojstvo. Ostatak dokumenta organiziran je na sljedeći način: ulomak 1.1 detaljno opisuje format ulazne i izlazne datoteke za zadatak A, dok ulomak 1.2 detaljno opisuje zadatak B te format ulazne i izlazne datoteke za zadatak B.

1.1 Zadatak A - NodeRank

Napomena: *Zadatak A nosi 60% bodova treće laboratorijske vježbe*.

U okviru ovog zadatka potrebno je izračunati rang za sve čvorove u zadanom usmjerenom grafu. Algoritam za izračunavanje ranga čvorova u grafu pseudokodom je opisan u sklopu predavanja te ga se može pronaći u materijalima na stranici predmeta (pdf, pptx). Bitno je naglasiti da ulazni graf neće sadržavati tzv. *dead end* čvorove!

Slijedi format ulazne datoteke:

$n \ \beta$
$A_{01} \ A_{02} \ A_{03} \ \ldots \ A_{0k_0}$
$A_{11} \ A_{12} \ A_{13} \ \ldots \ A_{1k_3}$
$A_{21} \ A_{22} \ A_{23} \ \ldots \ A_{2k_2}$
$\ldots$
$A_{n-11} \ A_{n-12} \ A_{n-13} \ \ldots \ A_{n-1k_n}$
q
$n_1 \ t_1$
$n_2 \ t_2$

$n_3\ t_3$
...
$n_q\ t_q$

U ulaznom zapisu, svaki redak završava znakom za kraj retka (**\n**). Linija 1. sadrži cijeli broj n i realni broj β koji su odvojeni prazninom. Broj n predstavlja broj čvorova u usmjerenom grafu i vrijedi $n \in [1,\ 10^5]$, dok β predstavlja vjerojatnost da slučajni šetač prilikom "šetnje" grafom slijedi bridove u grafu ($1 - \beta$ jest vjerojatnost teleportiranja).

Nakon 1. linije slijedi n linija koje opisuju bridove u grafu. U i-toj liniji navedeni su i prazninom razmaknuti redni brojevi čvorova za koje postoje bridovi iz čvora i počevši od odredišta A_{i1} do odredišta A_{ik_i}. Svaki redni broj odredišnog čvora A_{ij} jest u rasponu $[0,\ n-1]$ (dakle, koristi se 0-*based indexing* čvorova). Broj odredišnih čvorova za svaki čvor k_i kreće se u rasponu $[1,\ 15].$

Nakon n linija koje opisuju bridove između čvorova u grafu, slijedi jedna linija u kojoj je zapisan cijeli broj Q koji predstavlja broj upita na koje program treba odgovoriti. Slijedi Q linija gdje i-ta linija sadrži dva cijela broja n_i i t_i odvojena prazninom. Vrijedi: $Q \in [1,\ 10^5].$ Prvi broj n_i predstavlja indeks čvora i vrijedi: $n_i \in [0,\ n-1]$, a drugi broj t_i predstavlja redni broj iteracije algoritma *NodeRank* i vrijedi $t_i \in [1,\ 100]$.

Za svaki upit potrebno je u zasebnoj liniji ispisati vrijednost ranga traženog čvora n_i u traženoj iteraciji t_i.

Slijedi format izlazne datoteke:

$r_{n_1 t_1}$
$r_{n_2 t_2}$
$r_{n_3 t_3}$
...
$r_{n_q t_q}$

Traženu vrijednost potrebno je ispisati s preciznošću od **10** znamenki nakon decimalne točke!

Važne napomene:

- Vremensko ograničenje na izvođenje programa za bilo koju ulaznu definiciju automata jest 60 sekundi
- Ulazna točka za Java rješenja treba biti u **razredu NodeRank**, a ulazna točka u Python rješenja treba biti u datoteci **NodeRank.py**

1.1.1 Primjer za provjeru valjanosti

Na stranicama predmeta postavljen su primjeri ulaznih datoteka s pripadajućim očekivanim izlazima (*lab4A_primjeri.zip*). Preporučamo provjeru ispravnosti na temelju zadanih primjera prije same predaje vježbe na sustav *sprut*.

1.2 Zadatak - ClosestBlackNode

Napomena: *Zadatak B nosi 40% bodova treće laboratorijske vježbe*.

U okviru B zadatka ove vježbe zadan je neusmjereni graf G koji se sastoji od crnih i bijelih čvorova. Potrebno je za svaki čvor u grafu izračunati udaljenost do najbližeg crnog čvora te specificirati o kojem crnom čvoru se radi (ako za neki čvor postoji više crnih čvorova koji su mu najbliži, tada je potrebno ispisati crni čvor koji ima najmanji indeks).
Za svaki crni čvor udaljenost je po definiciji 0. Moguće da u ulaznom grafu postoje bijeli čvorovi iz kojih nije moguće doći do crnog čvora, što program treba ispisati u skladu s opisom koji je dan u specifikacije izlazne datoteke. Nadalje, za svaki čvor iz kojeg je moguće doći do nekog crnog čvora vrijedi da *maksimalna udaljenost* od tog čvora do nekog crnog čvora iznosi 10. Drugim rječima, ili je najbliži crni čvor udaljen za $dist \leq 10$, ili je crni čvor nedostupan.

Slijedi format ulazne datoteke:

$n \;\; e$
t_0
t_1
t_2
t_3
...
t_{n-1}
$s_0 \; d_0$
$s_1 \; d_1$
$s_2 \; d_2$
$s_3 \; d_3$
...
$s_{e-1} \; d_{e-1}$

U ulaznom zapisu, svaki redak završava znakom za kraj retka (**\n**). Linija 1. sadrži cijele broj n i e koji su odvojeni prazninom. Broj n predstavlja broj čvorova u neusmjerenom grafu i vrijedi $n \in [1,\ 10^5]$, dok e predstavlja ukupan broj bridova u grafu i vrijedi $e \leq 2.5 \cdot n$.

Nakon 1. linije slijedi n linija koje opisuju tipove čvorova u grafu. U i-toj liniji naveden je tip i-tog čvora t_i gdje je $t_i \in \{0, 1\}$ (čvor je crni ako je $t_i = 1$).

Nakon toga slijedi e linija koje opisiju bridova neusmjerenog grafa. U svakoj liniji prazninom su odvojeni dva broja s_i i d_i koji predstavljaju indekse čvorova između kojih postoji brid u neusmjerenom grafu i vrijedi $s_i, d_i \in [0, n-1]$ (dakle, koristi se *0-based indexing* čvorova u ulaznoj datoteci).

Program treba za svaki čvor u grafu ispisati *indeks* najbližeg crnog čvora i *udaljenost* do istog čvora.

Slijedi format izlazne datoteke:

$b_0\ dist_0$
$b_1\ dist_1$
$b_2\ dist_2$
$b_3\ dist_3$
...
$b_{n-1}\ dist_{n-1}$

Izlazna datoteka sastoji se od n linija. U i-toj liniji treba ispisati prazninom odvojena dva broja b_i i $dist_i$. Broj b_i jest indeks crnog čvora koji je najbliži i-tom čvoru, dok je $dist_i$ udaljenost i-tog čvora do crnog čvora b_i. Ako iz nekog čvora nije moguće doći do crnog čvora, tada program za taj čvor treba ispisati -1 -1 odvojene prazninom. Ako postoji više najbližih crnih čvorova, program treba ispisati onaj čvor koji ima najmanji indeks.

Važne napomene:

- Vremensko ograničenje na izvođenje programa za zadatak B za bilo koju ulaznu definiciju jest 10 sekundi
- Ulazna točka za Java rješenja treba biti u **razredu ClosestBlackNode**, a ulazna točka u Python rješenja treba biti u datoteci **ClosestBlackNode.py**

1.2.1 Primjeri za provjeru valjanosti

Na stranicama predmeta postavljeni su primjeri ulaznih datoteke s pripadajućim očekivanim izlazima (*lab4B_primjeri.zip*). Preporučamo provjeru ispravnosti na temelju zadanih primjera prije predaje vježbe na sustav *sprut*.