Distribuirani procesi **DRUGI KOLOKVIJ** **24.06.2026.**

Prezime i ime: _____

Zadatak 1

Na predavanju smo proučili klasu `ShortestPath.java`. Ona implementira algoritam za traženje najkraćih putova u usmjerenom grafu.

```
public class ShortestPath extends Process {
    int parent = -1;
    int cost = -1;
    int edgeWeight[] = null;

    public ShortestPath(Linker initComm, int initCost[]) {
        super(initComm);
        edgeWeight = initCost;
    }
    public synchronized void initiate() {
        if (myId == Symbols.coordinator) {
            parent = myId;
            cost = 0;
            sendToNeighbors("path", cost);
        }
    }
    public synchronized void handleMsg(Msg m, int src, String tag){
        if (tag.equals("path")) {
            int dist = m.getMessageInt();
            if ((parent == -1) || (dist + edgeWeight[src] < cost)) {
                parent = src;
                cost = dist + edgeWeight[src];
                System.out.println("New cost is " + cost);
                sendToNeighbors("path", cost);
            }
        }
    }
}
```

Objasnite vezu između tog algoritma i

- pojma difuzijskog računanja
- problema završetka rada.

Napišite sličnu klasu koja osim problema najkraćeg puta istovremeno rješava i problem *najduljeg puta*. Dakle za svaki čvor P_i treba pronaći najkraći i najdulji put od P_0 do P_i, s time da se duljina puta računa kao zbroj duljina lukova. Pretpostavljamo da su duljine lukova pozitivni cijeli brojevi.

Zaustavlja li se vaš algoritam za svaki usmjereni graf s pozitivnim duljinama lukova?

- Ako je odgovor „da“, dokažite da je to tako.
- Ako je odgovor „ne“, nacrtajte primjer grafa za koji se algoritam ne zaustavlja.

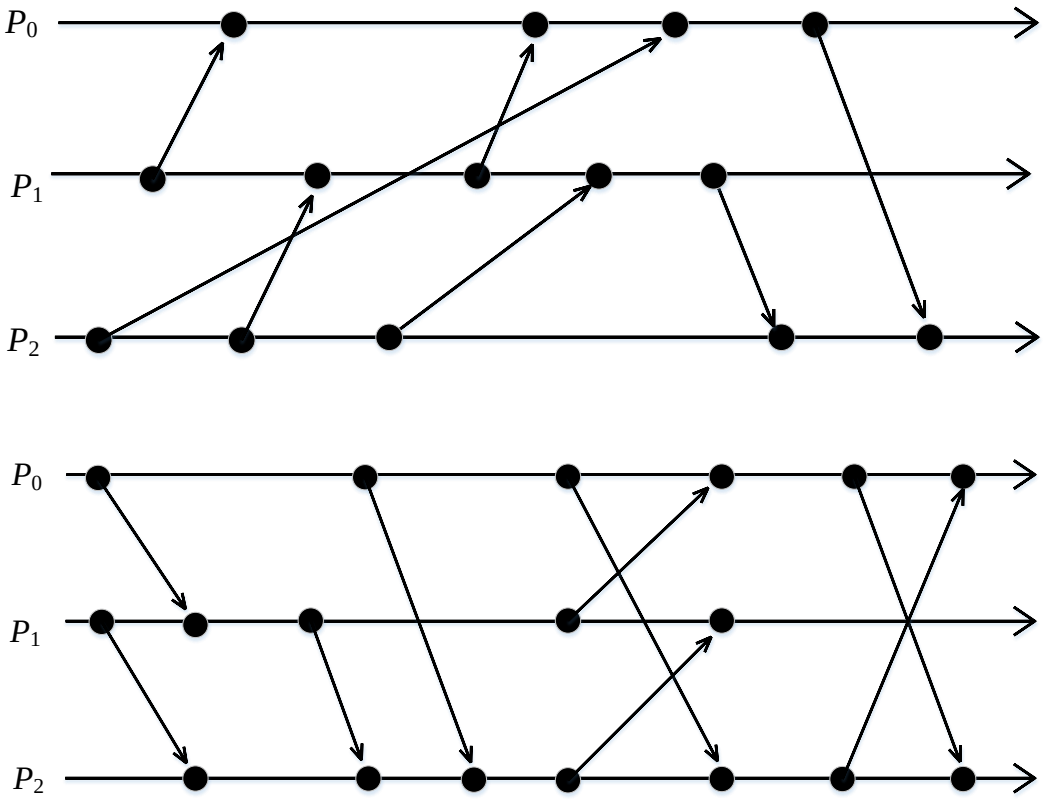
Distribuirani procesi **DRUGI KOLOKVIJ** **24.06.2026.**

Prezime i ime: _____

Zadatak 2

Promatramo distribuirano računanje zasnovano na skupu događaja E. Označimo s $<$ relaciju koja određuje redoslijed događaja unutar istog procesa, a s $\sim$ relaciju koja povezuje događaje slanja i primanja iste poruke. Definirajte pojam *FIFO* odnosno *kauzalnog* odnosno *sinkronog* uređaja: dakle pod kojim uvjetima kažemo da je promatrano računanje FIFO odnosno kauzalno odnosno sinkrono?

Nacrtani proces-vrijeme dijagrami prikazuju dva distribuirana računanja. Za svako od tih računanja posebno utvrdite zadovoljava li ono: FIFO, kauzalni, odnosno sinkroni uređaj poruka. Obrazložite vaše odgovore.



Nacrtajte dodatni dijagram koji prikazuje neko treće distribuirano računanje. Vaše računanje mora se sastojati od barem tri procesa i barem šest poruka. Pritom svaki proces mora poslati i primiti barem po jednu poruku, a uređaj poruka mora biti *sinkron* (dokažite da jest). Strelice za poruke ne smiju biti vertikalne (tj. treba predvidjeti da prijenos poruke u stvarnosti ipak troši neko vrijeme).

Distribuirani procesi **DRUGI KOLOKVIJ** **24.06.2026.**

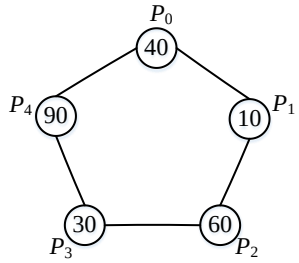
Prezime i ime: _____

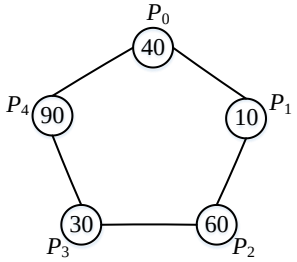
Zadatak 3

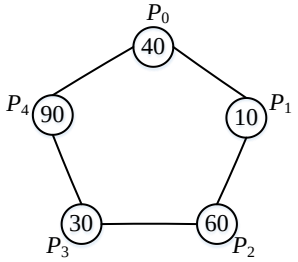
Objasnite zašto algoritam Chang-a i Roberts-a za izbor vođe u prstenu svoj izbor temelji na posebnim identifikatorima procesa, a ne na njihovim rednim brojevima 0, 1, 2, … N-1.

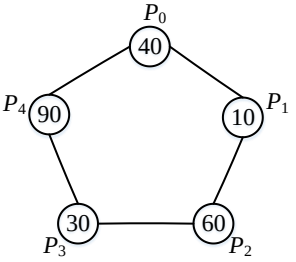
Pretpostavimo da u algoritmu Chang-a i Roberts-a točno 2 od N procesa u prstenu istovremeno pokreću postupak izbora vođe. Ocijenite broj poruka koje će se generirati u najboljem odnosno u najgorem slučaju.

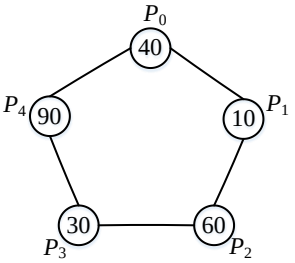
Služeći se nacrtanim dijagramima simulirajte rad algoritma Chang-a i Roberts-a na prstenu od 5 procesa. Niz dijagrama treba prikazati vremenski tijek algoritma. Svakom procesu upisan je identifikator koji služi kao kriterij za izbor vođe. Postupak izbora istovremeno pokreću procesi P_1 i P_3. Poruke putuju prstenom u smjeru kazaljke na satu. Na isti dijagram smijete ucrtati i više poruka ako se one odvijaju istovremeno. Dodajte još dijagrama ako je potrebno.

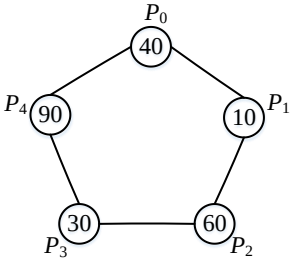


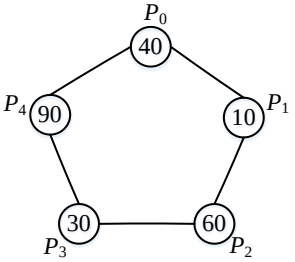


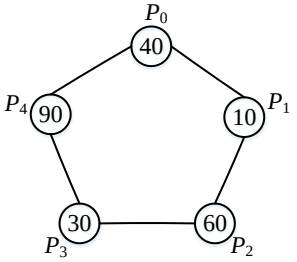












Distribuirani procesi **DRUGI KOLOKVIJ** **24.06.2026.**

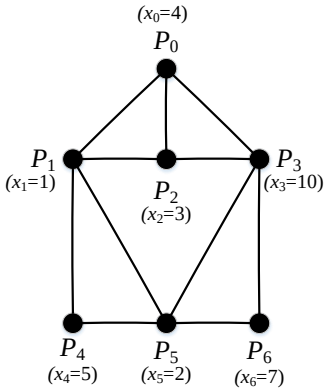
Prezime i ime: _____

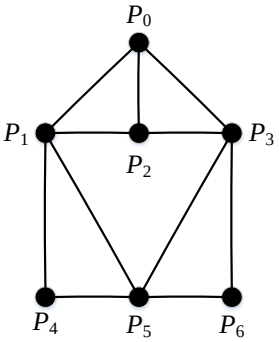
Zadatak 4

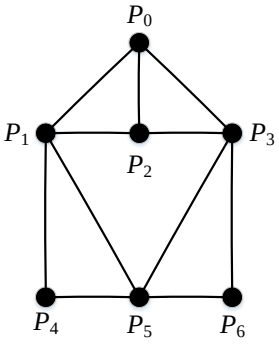
Napišite barem dvije (ekvivalentne) definicije razapinjućeg stabla neusmjerenog grafa. Objasnite zašto se razapinjuće stablo pojavljuje kao važan sastavni dio mnogih distribuiranih algoritama.

Služeći se nacrtanim dijagramima simulirajte rad algoritma za računanje globalne funkcije. Za zadani graf, globalna funkcija računa umnožak vrijednosti lokalnih varijabli x_0, x_1, … , x_6. Konkretne vrijednosti lokalnih varijabli vidljive su na prvom dijagramu.

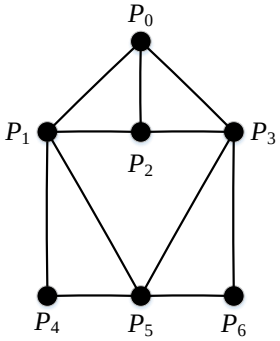
Započnite simulaciju tako da na prvi dijagram sami ucrtate jedno razapinjuće stablo. Pretpostavite da nakon izgradnje tog stabla svi procesi istovremeno pokreću algoritam računanja globalne funkcije. Zatim na ostale dijagrame ucrtajte poruke koje se prenose duž bridova stabla i kojima se realiziraju operacije *convergecast* i *broadcast* u sklopu algoritma. Niz dijagrama treba prikazati vremenski tijek algoritma. Na isti dijagram smijete ucrtati više poruka ako se one odvijaju (otprilike) istovremeno. Dodajte nove dijagrame ako je potrebno.

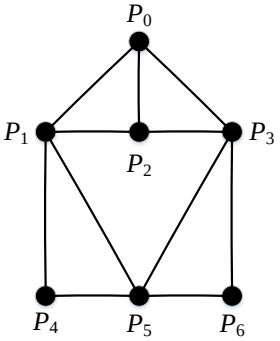


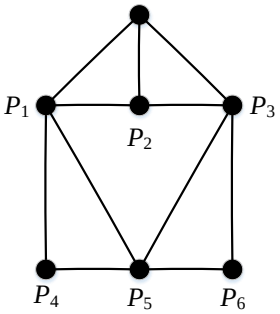




P_0







Distribuirani procesi **DRUGI KOLOKVIJ** **24.06.2026.**

Prezime i ime: _____

Zadatak 5

Definirajte pojam konsenzusa u *sinkronoj* mreži. Objasnite tri svojstva koja pri takvom konsenzusu moraju biti zadovoljena:

- usuglašenost
- valjanost
- konačnost.

Koji je smisao svojstva valjanosti, zašto je ono potrebno?

Može li u *sinkronoj* mreži jedan proces prepoznati da je drugi proces prestao raditi? Mijenja li se odgovor ako je mreža *asinkrona*? Obrazložite oba odgovora.

Skup od 30 procesa radi na *sinkronoj* mreži. Moguć je prestanak rada (kvar) najviše 10 od tih 30 procesa. Postoji li algoritam koji osigurava da se ispravni procesi usuglase oko vrijednosti jednog niza znakova (`String`-a) duljine 20? Ako je odgovor „da“, ugrubo opišite takav algoritam. U slučaju odgovora „ne“ dajte odgovarajuće obrazloženje. Mijenja li se odgovor ako je mreža *asinkrona*? Obrazložite.