



Powered by
Arizona State University

Strukture podataka i algoritmi
Seminarski rad

Prvi dio drugog domaćeg zadatka

Broj dosijea: **23/006**

Student: **Josif Vukićević**

Podgorica, septembar 2024. godine

1. Zbog čega se koriste binarna stabla u programiranju?

Binarna stabla se koriste u programiranju zbog svoje efikasnosti u organizaciji i pretraživanju podataka. Ona omogućavaju brz pristup, dodavanje i brisanje elemenata. Binarna stabla su osnova za implementaciju raznih algoritama poput pretraživanja, sortiranja i manipulacije nad podacima.

2. Objasniti zašto postoje različite vrste algoritama za sortiranje.

Postoje različite vrste algoritama za sortiranje jer svaki algoritam ima svoje prednosti i mane u smislu vremenske i prostorne složenosti. Primjer može biti da su neki algoritmi brži za manje skupove podataka, dok su drugi efikasnije nad većim skupovima podataka. Algoritmi sortiranja uključuju: bubble sort, insertion sort, selection sort, merge sort itd.

3. Na niz 10, 22, 13, -5, 12, -10, 11, 2 primijeniti algoritam merge sort i prikazati to slikovito.

[10, 22, 13, -5, 12, -10, 11, 2]

[10, 22, 13, -5] [12, -10, 11, 2]

[10, 22] [13, -5] [12, -10] [11, 2]

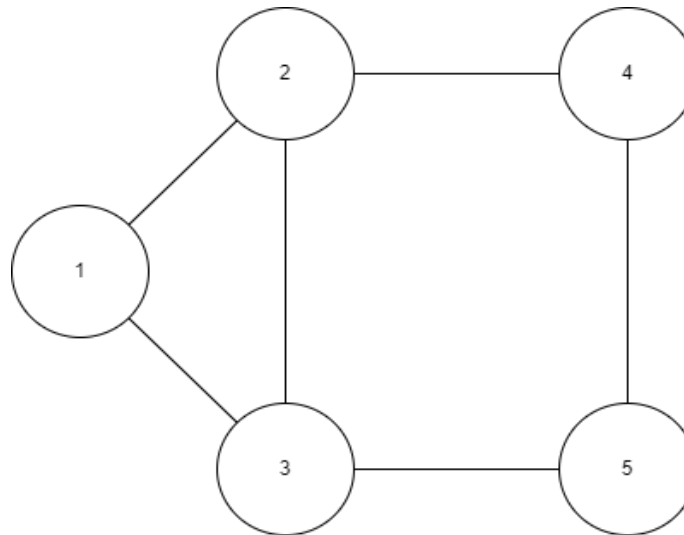
[10] [22] [13] [-5] [12] [-10] [11] [2]

[10, 22] [-5, 13] [-10, 12] [2, 11]

[-5, 10, 13, 22] [-10, 2, 11, 12]

[-10, -5, 2, 10, 11, 12, 13, 22]

4. Zbog čega se koriste grafovi? Kreirati primjer od 5 čvorova koji su nasumično povezani koji u sebi čuvaju brojeve. Skicirati BFS i DFS obilazak tih čvorova u grafu.



a.) BFS obilazak koristi red na sledeći način

1. Ubacivanje prvog cvora u red, i označavamo ga kao posjećenog

Red: 1

Posjećeno: 1

2. Sa cvora jedan posjećuju se cvorovi 2 i 3 koji će ubacuju u red i označavaju kao posjećene, a 1 se uklanja iz reda

Red: 2, 3

Posjećeno: 1, 2, 3

3. Sa cvora 2 se posjećuje cvor 4 koji se ubacuje u red i označava kao posjecen, dok se cvor 2 uklanja iz reda

Red: 3, 4

Posjećeno: 1, 2, 3, 4

4. Sa cvora 3 se posjećuje cvor 5 koji se ubacuje u red i označava kao posjecen, dok se cvor 3 uklanja iz reda

Red: 4, 5

Posjećeno: 1, 2, 3, 4, 5

5. Sa cvora 4 nema neposjecenih cvorova pa se on samo uklanja iz reda

Red: 5

Posjeceno: 1, 2, 3, 4, 5

6. Sa cvorom 5 je situacija ista kao i sa cvorom 4. Posto je red sada prazan BFS prolazak izgleda ovako:

Posjeceno: 1, 2, 3, 4, 5

b.) DFS obilazak koristi stek na sledeci nacin

1. Inicijalizacija steka i ubacivanje prvog cvora u posjecene i u stek

Stek: 1

Posjeceno: 1

2. Sa cvora jedan se posjecuje cvor 2, i ubacuje se u stek i posjecene

Stek: 1, 2

Posjeceno: 1, 2

3. Sa cvora 2 se posjecuje cvor 3, i ubacuje se u stek i posjecene

Stek: 1, 2, 3

Posjeceno: 1, 2, 3

4. Sa cvora 3 se posjecuje cvor 5 i dodaje se u stek i posjecene

Stek: 1, 2, 3, 5

Posjeceno: 1, 2, 3, 5

5. Sa cvora 5 se posjecuje cvor 4 i dodaje se u stek i posjecene

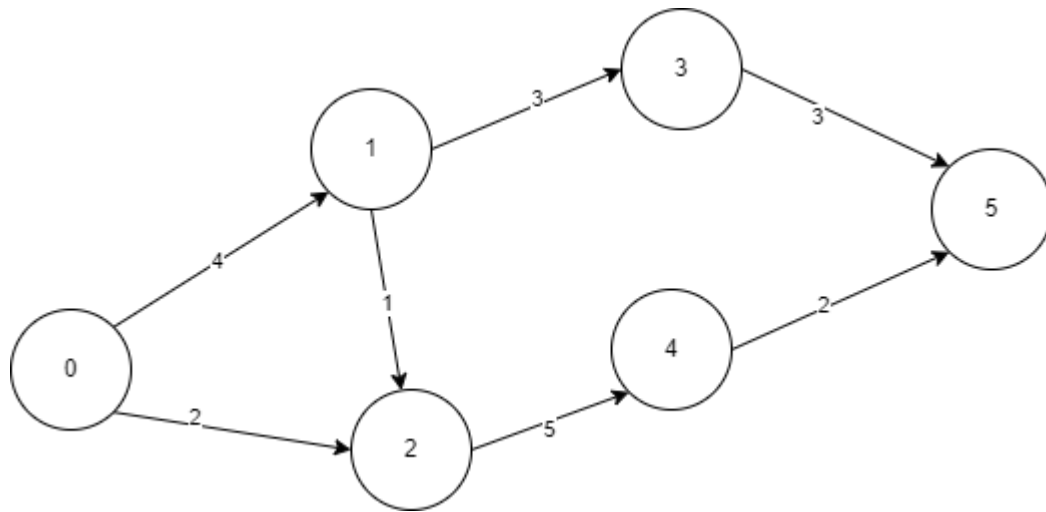
Stek: 1, 2, 3, 5, 4

Posjeceno: 1, 2, 3, 5, 4

6. Posto cvor 4 nema neposjecenih cvorova stek se polako prazni dok ne pronadje cvor sa neposjecenim susjedima. Takvih nema pa se stek prazni u potpunosti. DFS prolazak za ovaj stek izgleda ovako

Posjeceno: 1, 2, 3, 5, 4

5. Skicirati nalaženja najkraćeg puta za proizvoljan graf sa bar 5 čvorova koristeći algoritam Dijkstra.



Dijkstra algoritam doredjuje najkraci put do svih cvorova u grafu tako sto prati koji cvorovi su vec posjeceni i kolika je najmanja distanca do njih. Inicijalizacija neposjecenih cvorova i distanci izgleda ovako:

```

0 1 2 3 4 5
∞ ∞ ∞ ∞ ∞ ∞

```

Cvor sa najmanjom razdaljinom od cvora 0 je sam cvor 0, pa se on obiljezava posjecenim

```

0 1 2 3 4 5
0 ∞ ∞ ∞ ∞ ∞

```

Distance do susjednih cvorova su 4 do cvora 1 i 2 do cvora 2.

```

0 1 2 3 4 5
0 4 2 ∞ ∞ ∞

```

Cvor sa najmanjom distancom je sada cvor 2, razdaljina do njegovog susjednog cvora 4 je 5. Pa je razdaljina od cvora 2 do cvora 4 = 2 + 5 = 7

```

0 1 2 3 4 5
0 4 2 ∞ 7 ∞

```

Najblizi cvor je sada cvor 1, ovom cvoru je susjedan cvor 3 na razdaljini 3. Pa je razdaljina od cvora 1 do cvora 3 = 4 + 3 = 7

```

0 1 2 3 4 5
0 4 2 7 7 ∞

```

Cvorovi 3 i 4 su podjednako udaljeni od cvora 0 i susjedan im je cvor 5. Do cvora 5 je blize doci preko cvora 4 pa razdaljine izgledaju ovako

0 1 2 3 4 5

0 4 2 7 7 9

Distance do svakog cvora od cvora nula su sledece:

0 => 0

1 => 4

2 => 2

3 => 7

4 => 7

5 => 9