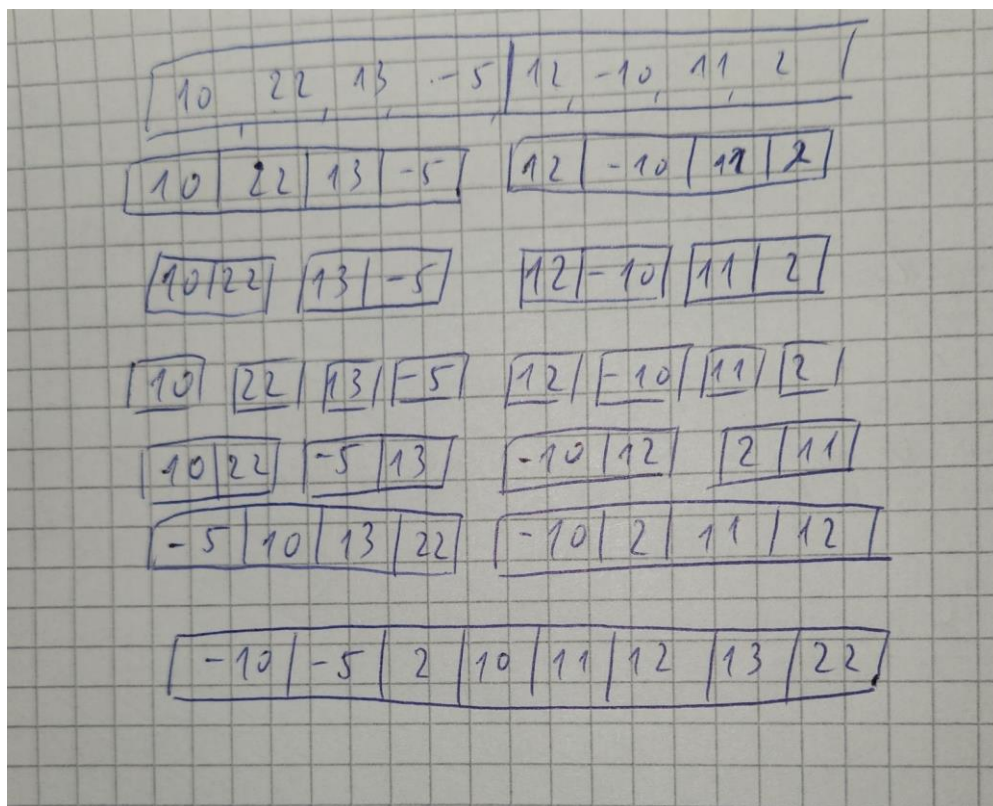


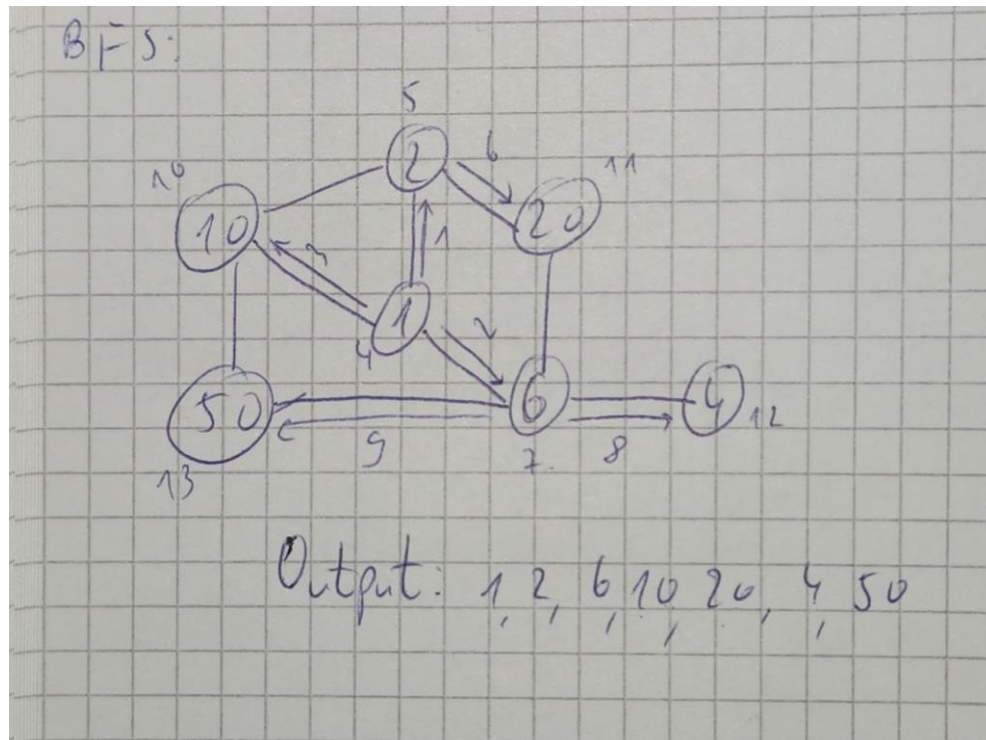
1. Zbog čega se koriste binarna stabla u programiranju?
 - Binarna stabla u programiranju se koriste jer su veoma efikasna u pronalaženju elemenata, zbog njihove strukture u kojoj su elementi sortirani pa je pretraga znatno olakšana jer se tačno zna sa koje strana datog elementa će se nalaziti traženi element.
2. Objasniti zašto postoje različite vrste algoritama za sortiranje.
 - Različite vrste algoritama za sortiranje se koriste jer neće svaki problem zahtijevati isti način rješavanja, svaki algoritam za sortiranje ima svoje benefite, ili kada je u pitanju vrijeme ili kada je u pitanju količina memorija koja se zauzima.
3. Na niz 10, 22, 13, -5, 12, -10, 11, 2 primijeniti algoritam merge sort i prikazati to slikovito.



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.

- Grafovi se koriste zbog međusobne veze između čvorova koja je veoma pogodna za modeliranje veza između nekih pojava koje želimo da predstavimo, takođe se koristi za optimizaciju ruta i predstavlja odličan način za analiziranje strukture mreže.

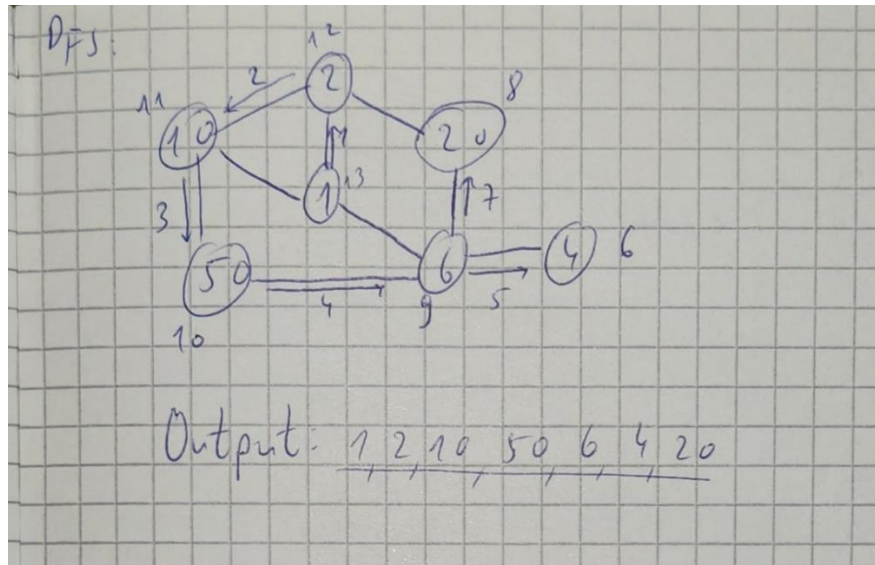
BFS:



Output : 1,2,6,10,20,4,50

- 1,2,3 – od čvora 1 obilazimo redom susjedne čvorove, prvo 2, pa 6, 10 i ubačamo ih u output
- 4 – popujemo čvor 1 jer smo sve njegove grane obišli
- 5,6 – popping čvor 2 i ujedno dodavanje 20 jer je susjedni čvor, 10 se već nalazi u outputu
- Prošli smo čvor 1 i 2, pa sad idemo dalje kroz output, na red je 6
- 7,8,9 – popping čvor 6 i dodavanje susjednih čvorova 4, pa 50 u output
- 10,11,12,13 – idemo dalje kroz output i vidimo da su grane svakog od preostalih čvorova već obišle pa ih samo popujemo (10 -> 20 -> 4 -> 50)

DFS:



Output: 1,2,10,50,6,4,20

- 1 – iz čvora 1 idemo ka čvoru 2 jer ima najmanju vrijednost u odnosu na ostale susjedne čvorove, dodajemo 2 u output i stack
- 2 – iz čvora 2 idemo u čvor 10 i dodajemo ga u output i u stack
- 3 – iz čvora 10 u čvor 50 i dodajemo ga u output i stack
- 4 – iz čvor 50 u susjedni čvor 6 koji nije obišen i dodajemo ga u output
- 5 – idemo do čvora 4, dodajemo ga u output na vrh stacka
- 6 - 4 nema dodatnih grana pa će da se poppuje iz stacka
- 7 – na čvor 6 je nadovezan i čvor 20 pa se on dodaje u output i na vrh stacka
- 8,9,10,11,12,13 – pošto 20 nema neobiđenih grana poppujemo ga sa vrha stacka, to isto radimo i sa svim preostalim elementima u stacku, 6, pa 50, pa 10, pa 2 i na kraju 1, jer je su svi čvorovi već pređeni

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

