

Strukture podataka i algoritmi 2
Januar, 2010

1. Radnja kompjuterske igre *IMI Strike* se odvija u sistemu soba jednog zamka. Svaka soba ima svoj jedinstveni broj, kao i četvora vrata (severna, istočna, južna i zapadna), kroz koja igrač može otići u neke od ostalih soba. Pojedina vrata u sobama mogu biti lažna, tako da ne vode nikuda. U svaku sobu igrač može doći samo iz jedne od ostalih soba. Naravno, kroz vrata na koja je ušao u neku sobu, igrač ne može otići u drugu sobu, već se samo može vratiti natrag u sobu iz koje je došao. U svakoj sobi se nalazi određeni broj dijamanta, koje igrač treba da sakupi.

Korišćenjem dinamičkih struktura podataka napraviti jezgro za ovu kompjutersku igru na sledeći način:

- a. Formirati odgovarajuću strukturu podataka koja će predstavljati sobu sa svim njenim podacima. **(3 poena)**
- b. Napisati funkciju *FindRoom* koja u izabranoj dinamičkoj strukturi pronalazi sobu sa zadatim brojem. **(5 poena)**
- c. Napisati funkciju *ReadConfiguration* koja iz tekstualne datoteke učitava podatke o sobama i povezuje ih na odgovarajući način unutar izabrane dinamičke strukture podataka. Svaka linija u ulaznoj datoteci predstavlja jednu sobu, tako što je prvo naveden jedinstveni broj te sobe, zatim broj sobe iz koje se u nju dolazi i redni broj vrata početne sobe kroz koja se dolazi u datu sobu (severna=0, istočna=1, južna=2, zapadna=3). Poslednji podatak u redu predstavlja broj dijamanta u toj sobi. Naravno, prva soba u datoteci nema prethodnika. Podrazumeva se da su podaci u ulaznoj datoteci korektni i da zadovoljavaju navedena pravila igre. **(10 poena)**
- d. Napisati funkciju *CollectDiamonds* koja određuje koliko se dijamanta može skupiti na putu od početne sobe do sobe sa zadatim brojem **(6 poena)** i koja štampa brojeve soba kroz koje se prolazi na tom putu, počevši od početne sobe. **(6 poena)**

Primer

Ulazna datoteka

```

1 -1 -1 2
2 1 0 5
3 2 0 7
4 3 3 1
5 2 1 3
6 5 1 0
7 6 2 6
  
```

Šematski prikaz soba

