

Laboratorijska vežba 2 – Algoritmi traženja

Student sa brojem indeksa k radi zadatak $(k \bmod 15)+1$.

Za izradu funkcija prepoznati i modifikovati (dodati nove argumente, promeniti vrednost koju funkcija vraća, izmeniti deo koda algoritma, dodati nove funkcije) odgovarajući algoritam obilaska stabla sa računskih vežbi. Sve probleme rešiti u okviru jednog obilaska grafa, ako nije drugačije navedeno.

Stablo traženja je stablo gde svaki put u stablu odgovara jednom putu u grafu. Unutar puta isti čvor se ne može pojaviti više puta.

Putevi koji nemaju nijedan zajednički čvor osim, eventualno, početnog i krajnjeg čvora su **disjunktni putevi**.

Podgrafovi koji nemaju nijednu granu koja povezuje bilo koji čvor prvog podgraфа sa bilo kojim čvorom drugog podgraфа su **disjunktni podgrafovi**.

1. Napisati funkciju koja određuje visinu stabla traženja (broj nivoa u stablu traženja), za algoritam obilaska grafa po širini, koje se formira za zadati polazni čvor i zadati graf.
2. Napisati funkciju koja određuje broj disjunktnih puteva između dva zadata čvora u grafu. Rešenje ne mora biti optimalno prema broju puteva. Dozvoljeno više puta pozvati algoritam obilaska grafa.
3. Napisati funkciju koja formira listu sastavljenu od elemenata (čvor, nivo) za sve čvorove grafa do kojih se može stići iz zadatog polaznog čvora, gde nivo predstavlja nivo stabla traženja na kome se nalazi čvor.
4. Napisati funkciju koja formira listu sastavljenu od elemenata (čvor, heuristika) za sve čvorove grafa do kojih se može stići od zadatog polaznog čvora. Heuristika za čvor se računa kao razlika broja čvorova u grafu i broja grana na putu od polaznog čvora do posmatranog čvora.
5. Napisati funkciju koja ispituje koliko disjunktnih podgrafova postoji u zadatom neorijentisanom grafu. Dozvoljeno više puta pozvati algoritam obilaska grafa.
6. Napisati funkciju koja formira stablo traženja za zadati graf, zadati polazni čvor i algoritam koji će se koristiti za obilazak stabla.
7. Napisati funkciju koja formira listu sastavljenu od elemenata (čvor, heuristika) za sve čvorove grafa do kojih se može stići od zadatog polaznog čvora. Heuristika za čvor se računa kao broj grana na putu od posmatranog do ciljnog čvora.

8. Napisati funkciju koja određuje put između zadatog polaznog i ciljnog čvora neorijentisanog grafa tako što istovremeno pokreće traženje po širini od polaznog i od ciljnog čvora. Traženje se završava kada se nađe prvi zajednički čvor za oba traženja.
9. Napisati funkciju koja određuje broj čvorova do kojih može da se dođe od zadatog čvora, tako da je dužina puta do čvora jednaka nekoj vrednosti. Običi samo neophodne čvorove.
10. Napisati funkciju koja pronalazi put u neorijentisanom grafu između dva zadata čvora, pri čemu put prolaze kroz treći zadati čvor.
11. Napisati funkciju koja određuje čvorove koje je moguće izbaciti iz neusmerenog grafa tako da graf i dalje ostane povezan.
12. Napisati funkciju koja pronalazi puteve zadate dužine u neorijentisanom grafu između dva zadata čvora.
13. Napisati funkciju koja određuje listu grana u grafu koje je neophodno obrisati da bi se ciklični graf transformisao u aciklični (da bi se u njemu eliminisali ciklusi).
14. Napisati funkciju koja određuje da li je orijentisani graf snažno povezani graf (da li postoji put između bilo koja dva čvora u grafu).
15. Napisati funkciju koja proverava da li graf sadrži Ojlerov put (put kojim se obilazi celokupan graf, uz obilazak svake grane tačno jednom).