

1.Naloga

a)

Uporabil bi dijkstra algoritem. Začnemo v točki S in nato pogledamo vse sosednje točke. Če ima sosednja točka že zapisano vrednost/utež in pot iz katere točke je prišla, potem primerjamo obe poti in če je manjša spremenimo utež in pot iz katere točke je prišla.

b)

Uporabil bi dijkstra algoritem od točke s do točke t in dijkstra algoritem od točke u do v. Če mi presek obeh množic poti vrne Vc notri potem algoritem vrne true.

2.Naloga

Iz začetne točke s bi izbral najmanjšo utež (ceno prenočitve), nato bi graf usmeril proti tej točki. Nato bi zopet iz začetne točke izbral drugo najmanjšo utež in bi zopet usmeril graf v to točko. To bi počel tako dolgo dokler ne zmanjka točk iz začetne točke. Nato bi šel na točko, ki sem jo že usmeril in bi ponovil postopek, tako kot pri začetni točki. Dobil bi usmerjen graf proti cilju t.

Ko bi se kolesarji odločali v katero mesto bodo šli, bodo pogledali utež (ceno prenočitve), to je npr. ko imaš na voljo 1 premik. Ko imaš na voljo 2 premika, bi za vsako naslednjo točko pogledal prav tako še vse njihove naslednje. In tista, ki vrne najmanjšo utež, tja bi šli.

3.Naloga

a)

Imamo desko neke dolžine in imamo vrednosti k. Desko odrežemo tam, kjer je k najbližje 0.50. Nato dobimo 2 deske in pogledamo deske in zareze na njih. Potem rekurzivno pogledamo zopet katera zares je najbližje polovici.

b)

Najbolj optimalno rešitev vidim v točki a, kjer je čas lahko logn v najbolšem primeru.