

Domača naloga #11: Uporabe DFS

Priprave na računalniške olimpijade 2018/19

Vid Kocijan
vid.kocijan@cs.ox.ac.uk

A. Fox and Names

Iščemo permutacijo abecede, v kateri bodo dana imena sortirana. Če primerjamo vsaki dve zaporedni imeni na seznamu, lahko dobimo množico pogojev, ki ji mora ta abeceda ustrezati oblike "črka a je pred črko b ". Primerjati moramo seveda zgolj prvi črki, v katerih se besedi razlikujeta. Da najdemo permutacijo, ki ustreza tem pogojem, uporabimo topološko sortiranje usmerjenem grafu, kjer vozlišča predstavljajo črke, usmerjene povezave pa urejenost.

B. Checkpoints

Opazimo, da ena postaja pokrije vsa križišča v povezani komponenti. V vsaki komponenti moramo preprosto najti najcenejšo lokacijo. V komponentah, kjer je takih lokacij več, izberemo poljubno. Število možnih postavitvev je produkt števila najcenejših lokacij v vsaki izmed komponent.

C. Bertown Roads

Če ima graf most, usmeritev, kot jo naloga zahteva, zagotovo ne obstaja. Kaj pa če mostov ni? Dokaz, da rešitev potem vedno obstaja prepustimo bralcu v premislek.

Obstaja več različnih postopkov, kako to rešitev najti. Ena možnost je, da z DFS najdemo poljuben cikel (npr. pot od vozlišča 1 do 1) in ga usmerimo v poljubno smer. Z malo premisleka opazimo, da je poljubna pot, ki se začne in konča na obiskanih vozliščih, med tem pa obiše zgolj prej neobiskana vozlišča, del veljavne rešitve. Postopek ponavljamo, dokler nismo v komponento dodali vseh vozlišč.

Dolžino kode lahko precej skrajšamo, če povezave spretno usmerjamo kar znotraj Tarjanovega algoritma za iskanje mostov.