Korešpondenčný seminár z programovania
XXXIII. ročník, 2015/16
Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

Návody k úlohám 2. kola letnej časti kategórie T

V kategórii T neuvádzame vzorové riešenia ale skôr návody na riešenie úloh. Ich cieľom je prezradiť vám hlavnú myšlienku riešenia, aby ste podľa návodu mohli riešenie domyslieť sami. Občas teda vynecháme niektoré drobné detaily, neuvádzame implementácie dátových štruktúr a nerozpisujeme niektoré kroky.

Neuvádzame ani vzorové programy, pretože chceme, aby ste po prečítaní návodu naprogramovali tie úlohy, ktoré ste nespravili počas trvania série. Keď si tieto riešenia sami naprogramujete, naučíte sa tým oveľa viac ako pozeraním sa na zdrojové kódy.

Implementácie všeobecne známych algoritmov a dátových štruktúr môžete nájsť vo vzorových riešeniach kategórií Z a O starších ročníkov KSP alebo aj na internete.

1. Tajná základňa

návod písal Buj
(max. 0 b za popis, 20 b za program)

Ak sa hrana a, b nachádza na vstupe (teda a a b nie sú spojené hranou), budeme hovoriť, že a je *antisused* b a tiež b je *antisused* a.

Jednoduché riešenie v čase $O(n^2)$: prehľadávame do šírky. Vždy, keď chceme nájsť susedov niektorého vrchola v, urobíme si pole *susedia* dĺžky n, v ktorom na začiatku zaznačíme, že každý vrchol ešte môže byť susedom v. Následne prechádzame zoznam antisusedov v. Ak b je antisused v, do poľa *susedia* si zaznačíme, že b nie je susedom v.

Na konci tohto procesu vieme z poľa *susedia* zostrojiť zoznam susedov v. To je jediné, čo potrebujeme pri prehľadávaní do šírky.

Jediné, čo nás delí od rýchleho riešenia, je nasledovné pozorovanie: Označme *visited* množinu vrcholov, ktoré sme doposiaľ v našom prehľadávaní navštívili. Potom nepotrebujeme pre žiaden z vrcholov vo *visited* vedieť, či susedí s v alebo nie – aj keby susedil, prehľadávanie v ňom nebude pokračovať, lebo ten vrchol bol navštívený už predtým.

Spracovanie vrcholu v bude teda prebiehať nasledovne – nech *notvisited* je štruktúra, ktorá obsahuje prehľadávaním dosiaľ nenavštívené vrcholy. Prejdeme zoznamom antisusedov v. Ak b je antisused v a zároveň b je v *notvisited*, tak si zaznačíme, že b stále nebude navštívené – vrchol b *označíme*. Na záver prejdeme zoznamom všetkých vrcholov v *notvisited*, a každý vrchol, ktorý sme neoznačili, pridáme do fronty a vyhodíme z *notvisited*.

Čitateľovi prenechávame na rozmyslenie, ako možno implementovať štruktúru *notvisited* a aká je časová zložitosť výsledného algoritmu.

2. Toľko voľného miesta!

návod písal Buj
(max. 0 b za popis, 20 b za program)

Ako vieme zo zadania, rozostavenie stoličiek je vyhovujúce práve vtedy, keď žiadne dve stoličky nie sú otočené oproti sebe. Takže zistiť, či je konkrétne rozostavenie vyhovujúce, vieme nasledovne:

Na začiatku každej stoličke A priradíme hodnotu 1. Následne, v smere, v ktorom je stolička A otočená, je práve jedna hrana oddeľujúca dve stoličky – stoličku A od niektorej inej stoličky B. Hrany oddeľujúce dve stoličky budeme volať *okraje*.

Presunieme hodnotu stoličky A na ten okraj, v ktorého smere je A otočená – hodnotu stoličky zmeníme na 0, a hodnotu okraju zvýšime o 1. Toto zopakujeme pre všetky stoličky. Potom rozostavenie vyhovuje práve vtedy, keď má každý okraj hodnotu najviac 1.

Zadanú úlohu potom vieme riešiť pomocou *minimum cost maximum flow* na nasledovne zostrojenom grafe: Vrcholmi grafu nech sú stoličky a okraje. Do každej stoličky vedie hrana zo zdroja (anglicky *source*) s kapacitou 1 a cenou 0. Z každej stoličky vedie hrana s kapacitou 1 do štyroch susedných okrajov. Cena každej z týchto hrán je určená tým, koľkokrát musíme stoličku otočiť, aby bola otočená smerom k tomu okraju. Z každého okraja vedie hrana s kapacitou 1 do výlevky (anglicky *sink*).

Rozmyslite si, prečo nám stačí hľadať cenovo najlacnejší maximálny tok – nemôže sa stať, že by tento tok niektorým hranám priradil iné hodnoty ako 0 alebo 1?

návod písal „“

3. Teatrálny Programovací Jazyk

(max. 0 b za popis, 20 b za program)

Pre nedostatok riešiteľov vzorové riešenie nemá písomnú podobu.

Chýba súbor vzor-prikl4.tex!!!

Chýba súbor vzor-prikl5.tex!!!