Pokyny

Písomka sa skladá zo štyroch (a bonusovej záverečnej) úloh, za ktoré viete získať dokopy 80 bodov.

Pri každej úlohe (ak nie je napísané inak) očakávam slovný popis jej riešenia. Tento popis by sa mal zamerať na hlavnú myšlienku riešenia, **odôvodnenie správnosti** a mal by obsahovať odhad časovej zložitosti.

Za riešenia je možné získavať aj čiastkové body a to buď za menej efektívne riešenia, alebo myšlienky smerujúce k správnemu riešeniu. Váš popis navyše môže obsahovať viacero rôzne efektívnych riešení, pričom body sa udeľujú za najlepšie správne riešenie. Ak si nie ste istý správnosťou vašeho riešenia, je **odporúčané** popísať aj pomalšie správne riešenie, aby ste v najhoršom prípade získali body aspoň zaň.

Ak používate algoritmus alebo postup priamo z prednášky, nemusíte ho bližšie popisovať, zamerajte sa na to, ako ho upravujete, poprípade aplikujete.

Počas písomky je **zakázané používať** akékoľvek písomné materiály či elektroniku. Písomka je closed-book, k dispozícii máte len papier a písacie potreby.

Odovzdávanie: Riešenie každej úlohy píšte na **samostatný papier**, riešenia sa budú zbierať po úlohách. Na každý papier svojho riešenia napíšte **svoje meno a číslo úlohy**, ak má riešenie niektorého príkladu viac ako jeden papier, je veľmi vhodné ich očíslovať.

Ak sa rozhodnete niektorú úlohu neriešiť, odovzdajte papier s vaším menom, číslom tejto úlohy a textom *NERIEŠIM*.

1. Tá o sumách podmnožín (25 bodov)

SUBSETSUM-1 – rozhodovací problém: pre zadanú množinu celých čísel A a číslo t rozhodnite, či existuje taká podmnožina A, ktorej súčet je práve hodnota t.

SUBSETSUM-2 – rozhodovací problém: pre zadané dve množiny celých čísel A a B rozhodnite, či existuje taká podmnožina A, ktorej súčet je prvkom množiny B.

(a) (3 body) Napíšte definíciu polynomiálnej redukcie rozhodovacieho problému X na rozhodovací problém Y.

(b) (7 bodov) **vymyslite a popíšte** polynomiálnu redukciu problému *SUBSETSUM-1* na problém *SUBSETSUM-2*.

(c) (15 bodov) **Vymyslite a popíšte** polynomiálnu redukciu problému *SUBSETSUM-2* na problém *SUBSETSUM-1*.

2. Tá o zvieratách (25 bodov)

Na izolovanom ostrove existuje veľmi jednoduchý živočíšny ekosystém so štyrmi druhmi zvierat – muchy, žaby, bociany a líšky. Keď ste prišli na ostrov, bolo na ňom 5 múch, 3 žaby, 17 bocianov a 2 líšky. Všimli ste si, že každý týždeň sa počet zvierat skokovo zmení a táto zmena sa riadi presnými pravidlami.

Nech m označuje počet múch, z počet žiab, b počet bocianov, l počet líšok a t počet uplynulých týždňov. Potom po uplynutí týždňa t budú nové počty zvierat m', z', b' a l' nasledovné:

- $m' = m + z + 20 - b$
- $z' = 3z + 3^t - 2m$
- $b' = 5t - 1$
- $l' = 2b + 2l + (t \bmod 2) \cdot 3^t + ((t + 1) \bmod 2) \cdot 2^t$

Teraz by vás zaujímalo, koľko zvierat bude na ostrove po práve t týždňoch. A keďže toto číslo môže byť fakt veľké, zaujíma vás to iba modulo prvočíslo $10^9 + 9$. (Navyše sa veľmi netrápte tým, že by počet zvierat klesol pod 0. Ak by sa také niečo náhodou stalo, príroda to zariadi tak, že v ekosystéme sa zjavia nové zvieratá tak, aby ich počet modulo sedel.)

Po prvom týždni na ostrove ($t = 1$) ste teda pozorovali 11 múch, 2 žaby, 4 bociany a 41 líšok. Po druhom týždni ($t = 2$) bude 29 múch, $10^9 + 9 - 7$ žiab, 9 bocianov a 94 líšok.

Vaše riešenie by malo okrem slovného popisu obsahovať aj **pseudokód** implementácie. Tento kód bude hodnotený zhruba polovicou bodov, pri jeho hodnotení budeme prihliadať na to, že ho píšete na papier – dôležitejšie ako preklepy a dokonalá syntax je správnosť hlavných výpočtov.

Až 10 bodov môžete získať za efektívne riešenie, ktoré predpokladá, že $t \leq 10^6$.

Až 20 bodov môžete získať za efektívne riešenie, ktoré predpokladá, že $t \leq 10^{12}$ a na ostrove žijú iba muchy, žaby a bociany. Počet líšok môžete v tomto riešení ignorovať.

Plných 25 bodov môžete získať za riešenie, ktoré predpokladá, že $t \leq 10^{12}$.

Čiastočné body sa dajú získať aj za popis efektívnych riešení, ktoré predpokladajú $t \leq 10^{12}$ a postupujú správnym spôsobom pri výpočte niektorých z hodnôt m', z', b' alebo l', hoci ostatné z týchto hodnôt nevedia počítať správne alebo dostatočne efektívne.

3. Tá s Kruskalom (25 bodov)

Máme graf, v ktorom chceme postupne obsadiť **každý vrchol** (nie nutne každú hranu) našimi figúrkami. Na začiatku nie je obsadený žiaden vrchol ani hrana a v grafe je nula našich figúrok.

Vrcholy aj hrany sa obsadia tak, že na ne dostaneme dostatočný počet našich figúrok. Keď niektorú časť grafu obsadíme, patrí nám už do konca a to bez ohľadu na to, či odtiaľ naše figúrky presunieme.

Za cieľom obsadzovania vieme robiť nasledovné úkony:

- Obsadiť vrchol v, v ktorom sa nachádza aspoň a_v figúrok.
- Obsadiť hranu e, ak je vo vrcholoch, ktoré spája dokopy aspoň b_e figúrok.
- Zaplatiť c_v peňazí a pridať do vrcholu v jednu figúrku.
- Presunúť figúrku po *obsadenej* hrane (a to bez ohľadu na to, či sú obsadené koncové vrcholy tejto hrany).

Na vstupe dostanete popis grafu. Pre každý vrchol sú zadané hodnoty a_v a c_v (ktoré sa môžu pre rôzne vrcholy líšiť), a každá hrana je popísaná vrcholmi, ktoré spája a hodnotou b_e.

Obsadiť každý vrchol grafu je pomerne jednoduché, stačí ak do každého vrcholu kúpime dostatočný počet figúrok. To by nás však stálo veľa peňazí, lepšie (asi) teda bude obsadiť aj nejaké hrany a figúrky postupne popresúvať. Zistite, ako najlacnejšie vieme obsadiť všetky vrcholy grafu.

(a) (8 bodov) Nech vieme, ktoré hrany budú obsadené v optimálnom riešení. Navrhnite spôsob, akým vypočítať najmenšiu cenu takéhoto riešenia, teda najmenšiu cenu potrebnú na obsadenie všetkých vrcholov a práve týchto hrán.

(b) (4 body) Dokážte, že ak sa rozhodneme obsadiť nejakú hranu, existuje optimáne riešenie, ktoré ju obsadí iba z jednej strany – teda všetkých b_v figúrok potrebných na jej obsadenie dáme do jedného z jej vrcholov.

(c) (13 bodov) Optimálne riešenie tejto úlohy využíva upravený Kruskalov algoritmus. Hrany prechádza v poradí podľa hodnoty b_v, pričom si udržiava aktuálne komponenty „súvislosti“. Pre každý komponent súvislosti vie, ako najlacnejšie sa dajú obsadiť všetky vrcholy v ňom. Domyslite, popíšte a poriadne zdôvodnite všetky detaily tohto riešenia.

Sústreďte sa najmä na popis toho, čo spraví algoritmus pri pridávaní novej hrany. Ako vypočíta najlacnejšie obsadenie tohto nového komponentu? Aké informácie si musí algoritmus pamätať pre každý komponent? A prečo je toto riešenie správne a naozaj vypočíta najlacnejšiu cenu obsadenia?

4. Tá na doplnenie do 80 (5 bodov)

Pre každý z nasledujúcich výrokov určite či je pravdivý a svoje tvrdenie odôvodnite. Za správnu pravdivosť dostanete 0.5 bodu, zvyšné 2 za správne odôvodnenie.

(a) Po obvode kruhu sme napísali n písmen. Zaujíma nás, či niektorých m písmen, ktoré ležia na kruhu za sebou, tvorí hľadaný reťazec P. Predpokladáme, že $m \leq n$. Vieme túto úlohu riešiť v časovej zložitosti $O(n + m)$?

(b) Nech P je najkratšia cesta medzi vrcholmi a a b v ohodnotenom grafe G. Ak z grafu G vytvoríme graf G' tak, že váhu každej hrany zdvojnásobíme, P bude najkratšou cestou medzi a a b aj v G'.

5. Tá o ankete (2 body)

Túto úlohu odovzdávajte spoločne na papieri s úlohou 4.

Čestne prehlasujete, že po príchode domov vyplníte anketu k predmetu Tvorba efektívnych algoritmov, optimálne aj so slovným komentárom. Následne ku koncu skúškového anketu aj odošlete (či tam vyplníte zvyšok je na vás).

Ďakujem pekne za spätnú väzbu :)