72. ročník Matematické olympiády – 2022/2023

Úlohy ústředního kola kategorie P – 1. soutěžní den

Na řešení úloh máte 4.5 hodiny čistého času. Řešení každé úlohy pište na samostatný list papíru. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb (tzn. knihy, kalkulačky, mobily, apod.).

Řešení každé úlohy má obsahovat:

- *Popis řešení,* to znamená slovní popis principu zvoleného algoritmu, *argumenty zdůvodňující jeho správnost* (případně důkaz správnosti algoritmu), *diskusi o efektivitě* vašeho řešení (časová a paměťová složitost; to se netýká úlohy P-III-3). Slovní popis řešení musí být jasný a srozumitelný i bez nahlédnutí do samotného zápisu algoritmu (do programu). Není možné odkazovat se na vaše řešení úloh z předchozích kol, opravovatelé je nemusí mít k dispozici.
- Doporučujeme uvést *zápis algoritmu* v nějakém dostatečně srozumitelném pseudokódu (případně v programovacím jazyce C/C++, Python nebo podobném). Nemusíte detailně popisovat jednoduché operace jako vstupy, výstupy, implementaci jednoduchých matematických vztahů, vyhledávání v poli, třídění apod. Zápis algoritmu sice není povinnou součástí řešení, ale při nejasností v popisu řešení může opravovatelům pomoci s pochopením algoritmu.

Za každou úlohu můžete získat maximálně 10 bodů. Hodnotí se nejen správnost řešení, ale také kvalita jeho popisu a efektivita zvoleného algoritmu. Algoritmy posuzujeme podle jejich časové složitosti, tzn. závislosti doby výpočtu na velikosti vstupních dat. Záleží přitom pouze na řádové rychlosti růstu této funkce. V zadání každé úlohy najdete přibližné limity na velikost vstupních dat. Efektivním vyřešením úlohy rozumíme to, že váš program spuštěný s takovými daty na současném běžném počítači dokončí výpočet během několika sekund.

P-III-1 Raut

Kvalita konference se pozná především podle množství občerstvení, které je o přestávkách k dispozici. Z Vaška se stal slavný vědec a jako takový začal dostávat pozvánky na velice kvalitní konference. Vašek by samozřejmě nejraději o přestávkách snědl všechno občerstvení, ale toho je příliš mnoho. Rozhodl se tedy, že se pokusí sníst co nejvíce co nejlepších zákusků během přestávky a jeden zákusek si pak ještě vezme s sebou na další přednášku. Pomůžete mu naplánovat, které zákusky jíst?

Soutěžní úloha

Na stole je n dobrot, i-tá z nich má *mňamóznost* m_i a Vaškovi by trvalo t_i vteřin ji sníst. Přestávka končí za T vteřin. Pomozte Vaškovi rozhodnout, které dobroty má sníst ještě během přestávky a kterou si s sebou má vzít na přednášku tak, aby měly dohromady co největší součet mňamóznosti. Formálně řečeno, najděte množinu $S \subseteq \{1, \ldots, n\}$ a číslo $p \in \{1, \ldots, n\}$ takové, že $\sum_{i \in S} t_i \leq T$, $p \notin S$ a součet $m_p + \sum_{i \in S} m_i$ je co největší.

Formát vstupu

Na prvním řádku dostanete dvě mezerou oddělená přirozená čísla n, T. Na druhém řádku dostanete n mezerou oddělených přirozených čísel $m_1, \ldots, m_n$. Na třetím řádku dostanete n mezerou oddělených přirozených čísel $t_1, \ldots, t_n$.

Formát výstupu

Nechť množina S a index p dávají největší možný součet mňamóznností. Potom na první řádek výstupu vypište prvky S v libovolném pořadí a na druhý řádek vypište index p. První z těchto řádků může potenciálně být prázdný.

Pokud je správných řešení vícero, vypište libovolné z nich.

Částečné body lze získat za řešení, které pouze na jediný řádek výstupu napíše největší možný součet mňamóznností $m_p + \sum_{i \in S} m_i$, kterého může Vašek dosáhnout. (Viz sekce Bodování.)

Příklady

Vstup:	*Výstup:*
4 6 3 4 1 5 5 3 2 4	3 4 2

Největší součet mňamóznosti zákusků, které Vašek sní, je 10, čehož může Vašek dosáhnout snědením třetího a čtvrtého zákusku o přestávce a tím, že si na přednášku vezme druhý zákusek. První zákusek musí nechat nesnědený – pokud by jej snědl o přestávce, nemohl by sníst nic dalšího, což se mu určitě nevyplatí. Na další přednášku si ale může vždy vzít lepší zákusek. Dalším vyhovujícím řešením by bylo sníst druhý a třetí zákusek hned a čtvrtý o přednášce.

Vstup:	*Výstup:*
3 4 2 2 1 5 3 100	2 1

Vašek může sníst druhý zákusek o přestávce a první si vzít na přednášku. Poslední zákusek, který by jedl velmi dlouho, nebude jíst vůbec.

Vstup:	*Výstup:*
3 20 1 4 6 5 3 8	2 3 1

Vašek si první dortík vezme na přednášku.

Vstup:	*Výstup:*
1 20 1 21	1

Vašek nestihne sníst dortík o přestávce, může si jej ale vzít na přednášku.

Bodování

Vždy bude platit $1 \leq n \leq 10^4$, $1 \leq T \leq 10^4$ a pro všechna $1 \leq i \leq n$ bude platit $1 \leq m_i \leq 10^9$ a $1 \leq t_i \leq 10^9$.

Pokud úlohu nevyřešíte plně, ale vypíšete pouze optimální hodnotu $m_p + \sum_{i \in S} m_i$, dostanete počet bodů, který je uvedený v následujícím seznamu v závorkách.

- Až 10 bodů (popř. až 7 bodů) dostanete za řešení, která efektivně řeší vstupy bez dalších omezení.
- Až 6 bodů (popř. až 5 bodů) dostanete za řešení, která navíc předpokládají, že $t_i = m_i$.
- Až 4 body (popř. až 3 body) dostanete řešení efektivní pro vstupy, kde $n \leq 100$.

P-III-2 Lezení

Na Olympijských hrách v Paříži 2024 se v soutěži v lezecké kombinaci bude závodit ve dvou disciplínách – v boulderingu a v lezení na obtížnost. Shodou nešťastných okolností a donebevolajících pochybení byl za Českou republiku místo Adama Ondry nominovaný náš organizátor Ondra, který vůbec boulderovat ani lézt na obtížnost neumí. Aby si tam neuřízl ostudu, Ondra se rozhodl, že se do závodu naučí boulderovat i lézt na obtížnost stejně dobře jako Adam Ondra. Pomůžete mu vybrat, kam chodit trénovat?

Soutěžní úloha

V Praze je n lezeckých stěn očíslovaných $1, \ldots, n$. Pokud Ondra stráví hodinu na stěně číslo i, zlepší se v boulderingu o b_i jednotek a v lezení na obtížnost o ℓ_i jednotek. Ondrovy schopnosti v boulderingu i v lezení na obtížnost jsou momentálně 0 jednotek, Adam Ondra má B jednotek schopnosti v boulderingu a L jednotek schopnosti v lezení na obtížnost. Poraďte Ondrovi, na kterých stěnách má strávit kolik času, aby byl alespoň tak dobrý jako Adam Ondra v boulderingu i v lezení na obtížnost. Protože Ondra musí zároveň organizovat olympiádu, potřebuje na stěnách dohromady strávit co nejmenší množství času. (Upozornění: časy, které Ondra stráví na stěnách, nemusí být celočíselné.)

Formát vstupu

Na prvním řádku dostanete tři mezerou oddělená přirozená čísla n, B a L. Na druhém řádku dostanete n mezerou oddělených přirozených čísel $b_1, \ldots, b_n$. Na třetím řádku dostanete n mezerou oddělených přirozených čísel $\ell_1, \ldots, \ell_n$.

Formát výstupu

Na jediný řádek výstupu vypište jedno nezáporné reálné číslo t, která splňuje následující podmínky:

- $t = t_1 + \cdots + t_n$, kde $t_1, \ldots, t_n$ jsou nezáporná reálná čísla.
- Pokud Ondra stráví na první stěně t_1 hodin, na druhé stěně t_2 hodin a tak dále, tak na konci bude mít alespoň B jednotek schopnosti v boulderingu a alespoň L jednotek schopnosti v lezení na obtížnost.
- Číslo t je nejmenší, které splňuje tyto podmínky. Můžete předpokládat, že takové nejmenší t vždy existuje (lze to za podmínek úlohy dokázat).

Jelikož jde o teoretickou úlohu, nemusíte brát v potaz zaokrouhlovací chyby při práci s neceločíselnými proměnnými.

Příklady

Vstup:	*Výstup:*
2 3 3 1 2 2 1	2.0

V tomto případě stráví Ondra hodinu na každé stěně.

Vstup:	*Výstup:*
2 4 3 6 2 2 4	1.0

Tentokrát je pro Ondru nejlepší strávit na každé ze stěn půl hodiny.

Vstup:	*Výstup:*
3 6 6 1 3 5 5 3 1	2.0

V tomto případě může Ondra například na každé stěně strávit 40 minut.

Vstup:	*Výstup:*
2 3 3 1 100 1 6	0.5

Ondra svého cíle může dosáhnout tak, že zajde na půl hodiny na druhou stěnu.

Bodování

Vždy bude platit $1 \leq n \leq 10^5$, $1 \leq B, L \leq 10^9$ a pro všechna $1 \leq i \leq n$ bude platit $1 \leq b_i, \ell_i \leq 10^9$. Můžete také předpokládat, že žádné dvě stěny nemají stejné parametry, tedy pokud $b_i = b_j$ a $\ell_i = \ell_j$, pak $i = j$.

Tato úloha má čtyři podúlohy, v nichž mohou platit nějaké podmínky navíc. Tabulka níže tyto podúlohy popisuje a určuje, kolik bodů za ně lze získat.

body	*podmínky*
2	$n = 2$
3	$n = 1\,000$
2	$1 \leq b_i, \ell_i \leq 1\,000$
3	žádná další omezení

P-III-3 Hvězdní věštci

K této úloze se vztahuje studijní text uvedený na následujících stranách, který je totožný se studijním textem ze zadání domácího a krajského kola.

a) *(4 body)*

Císař by rád propojil Hvězdné impérium systémem mezihvězdných dálnic. Chce ale, aby byly splněny následující požadavky:

- Každá dálnice spojuje dva sousední systémy (tedy ty, které spolu umí přímo komunikovat).
- Mezi každými dvěma systémy se dá dojet po dálnicích.
- Dálnic je co nejméně, tj. pokud bychom nějakou dálnici odstranili, mezi nějakými dvěma systémy by se nedalo dostat (formálně řečeno, systém dálnic tvoří strom).
- Pro zjednodušení dopravního značení smí z každého systému vycházet nejvýše tři dálnice.

Navrhněte algoritmus, který odpoví ANO, právě když je možné postavit systém mezihvězdných dálnic splňující tyto podmínky.

b) *(6 bodů)*

Jako alternativu císařští poradci navrhli prostě vybudovat mezihvězdnou okresku mezi každými dvěma sousedními systémy. Problém je, že k dispozici jsou peníze pouze na vybudování M okresek (toto číslo $M \leq \binom{N}{2}$ je známo ve všech hvězdných systémech). Navrhněte algoritmus, který odpoví ANO, právě když počet neuspořádaných dvojic sousedních systémů je menší nebo roven M.

Připomínáme, že při hodnocení této úlohy *nezáleží na časové a paměťové složitosti* vašeho řešení, pouze na jeho správnosti, hodnotě K a vašem zdůvodnění.

Studijní text

Vědci Hvězdného impéria vyřešili již téměř všechny problémy, které říši trápí. Díky vynálezu hyperkvantových počítačů jsou například schopni provést libovolný výpočet v konstantním čase. Velkým problémem ale zůstává komunikace mezi hvězdnými systémy. Přímo poslat zprávu lze pouze do blízkých systémů, kterým budeme říkat *sousední;* i pro ně ale doručení zprávy trvá jeden rok.

Mezi každými dvěma systémy Hvězdného impéria se sice lze dostat přes posloupnost sousedních systémů (tj. graf, jehož hrany tvoří sousední systémy, je souvislý), ale některé systémy jsou tak daleko od sebe, že zaslání zprávy mezi nimi by trvalo několik tisíciletí, a je tedy zcela nepraktické. Tradičně tak jediným globálním komunikačním kanálem byla císařova telepatická schopnost komukoliv okamžitě předat jakoukoliv informaci. Tento kanál je bohužel jednosměrný, odpovědět obdobně císaři nelze.

Nedávno se podařilo najít alespoň částečné řešení a vytvořit bezdrátovou síť 115. generace, kterou každý může poslat zprávu všem v celém vesmíru a tato zpráva je doručena okamžitě. Komplikaci je ale nízká propustnost této sítě, konkrétně jeden bit za rok (pokud se více osob pokusí v ten samý rok síť použít, doručena je jen první z odeslaných zpráv). K řešení různých otázek proto Hvězdné impérium používá následující protokol, kombinující moderní techniku s tradičními metodami:

- Císař všem sdělí počet N hvězdných systémů, každému hvězdnému systému přidělí číslo od 0 do $N-1$ jako jednoznačný identifikátor a sdělí, jaký (všichni stejný) algoritmus mají použít k řešení dané otázky.
- Vládce každého hvězdného systému zajde do místní věštírny, kde mu sdělí K-bitový řetězec R (ne nutně ve všech hvězdných systémech stejný).
- Na základě R a předepsaného algoritmu pošle všem sousedním (tj. nejvýše jeden světelný rok vzdáleným) systémům zprávu.
- Poté, co přijme zprávy od všech sousedních systémů, na jejich základě dle předepsaného algoritmu rozhodne, zda odpověď na otázku je ANO nebo NE. Je-li odpověď NE, pošle všem bezdrátovou sítí jeden bit a popraví věštce.
- Pokud nikdo po jednom roce tento bit neposlal, odpověď je ANO.

Funkčnost tohoto systému je zaručena tím, že, jak se obecně ví, věštci disponují magickou schopností si zachránit život, pokud je to jen možné. Existuje-li tedy nějaká volba řetězců R taková, že ve všech systémech daný algoritmus dospěje k odpovědi ANO, věštci ji dokážou najít.

Věštci si účtují horentní sumy za každý vyvěštěný bit, naším cílem je tedy najít algoritmus s co nejmenším K.

Příklad 1: Tři týmy

Chceme vědět, zda se hvězdné systémy mohou rozdělit do tří týmů tak, aby žádné dva systémy patřící do jednoho týmu nesousedili. Vládce každého systému si tedy nechá vyvěštit dva bity 01, 10 nebo 11 (pokud by věštec vyvěštil 00, rovnou ho

popraví), udávající číslo týmu, do kterého bude jeho systém patřit. Toto číslo pak pošle všem sousedním systémům. Odpověď je NE, pokud od některého ze sousedních systémů dostane stejné číslo.

Lze-li systémy do tří týmů rozdělit, věštci mohou pro každého uhodnout číslo jeho týmu, všichni přežijí a víme, že na otázku je odpověď ANO. Pokud to není možné, nějací dva sousední věštci musí zvolit stejné číslo, budou popraveni a odpověď je NE.

Popsaná strategie používá $K = 2$.

Příklad 2: Univerzální řešení

Libovolnou úlohu můžeme vyřešit tak, že si od věštce necháme vyvěštit mapu celého hvězdného impéria, tj. matici o rozměrech $N \times N$, kde v i-tém řádku a j-tém sloupci je 1, právě když hvězdné systémy i a j sousedí. Tuto mapu pošleme sousedům, a pokud od některého souseda dostaneme jinou mapu, popravíme věštce. Jelikož je Hvězdné impérium souvislé, tímto zaručíme, že věštci všem musí vyvěštit stejnou mapu. Také všem sousedům pošleme svůj identifikátor a na základě identifikátorů, které obdržíme od sousedů, ověříme, že jsou v mapě správně uvedeni naši sousedé (pokud ne, věštce opět popravíme). Tím zaručíme, že vyvěštěná mapa správně popisuje celé Hvězdné impérium.

Jelikož máme k dispozici neomezenou výpočetní sílu, každý systém zvlášť pak může úlohu vyřešit a odpovědět ANO nebo NE. Tato strategie používá $K = N^2$; obecně proto získáte body pouze za řešení, pro něž je K výrazně menší než N^2.

Popis algoritmu

Algoritmus můžete buď dostatečně přesně popsat, nebo zapsat ve vhodném pseudokódu. V tomto pseudokódu můžete používat:

- Počet systémů N a identifikátor A aktuálního systému, což je přirozené číslo mezi 0 a $N - 1$.
- Proměnnou R libovolného typu, v níž je uložena věštba.
- Přirozené číslo D, udávající počet sousedních systémů.
- Pole `send` obsahující D prvků libovolného typu, na i-tou pozici v tomto poli ukládáte zprávu, kterou pošlete i-tému ze sousedních systémů.
- Pole `receive` obsahující D prvků stejného typu, na i-té pozici naleznete zprávu, kterou obdržíte od i-tého ze sousedních systémů. Toto pole smíte číst až poté, co provedete všechny zápisy do pole `send`.
- Váš program na konci musí vrátit odpověď ANO nebo NE.

Aby bylo zřejmé, kolik bitů věštby používáte, můžete v definici typu R použít například typ `unsigned(s)`, označující s-bitové přirozené číslo.

Ve svých řešeních neurčujte paměťovou a časovou složitost, na té nám nezáleží. Vaše řešení bude hodnoceno pouze na základě jeho správnosti, zdůvodnění a hodnoty K.

Příklad 3: Cesta

Nechť $N \geq 2$. Chceme rozhodnout, zda Hvězdné impérium tvoří cestu, tj. zda lze systémy seřadit do posloupnosti $s_0, s_1, \ldots, s_{N-1}$ tak, aby systém s_0 sousedil pouze se systémem s_1, s_{N-1} pouze s s_{N-2} a pro $i = 1, \ldots, N-2$ systém s_i sousedil právě se systémy s_{i-1} a s_{i+1}.

Má-li některý systém více než dva sousedy, odpověď je zjevně NE. Pokud všechny mají nejvýše dva sousedy, pak jelikož je Hvězdné impérium souvislé, musíme pouze rozhodnout, zda tvoří cestu (tj. právě dva systémy mají jen jednoho souseda) nebo kružnici (všechny systémy mají dva sousedy). To můžeme rozhodnout například tak, že nám věštec řekne naši pozici p v posloupnosti a to, který z našich dvou sousedů je před námi, na základě těchto informací oběma sousedům pošleme jejich pozici v posloupnosti (jednomu z nich $p-1$ a druhému $p+1$) a odpovíme ANO jen tehdy, když od obou sousedů dostaneme p. Systémy s pouze jedním sousedem postupují obdobně, jen ještě ověří, zda jejich pozice je 0 nebo $N-1$.

V pseudokódu můžeme algoritmus zapsat následovně (funkce `ceil` značí horní celou část a `log` je dvojkový logaritmus):

```
typedef unsigned(ceil(log(N))) pozice;

struct
{
  pozice p;
  unsigned(1) pred;
} R;

pozice send[D], receive[D];

if (D > 2)
  return NE;

if (D == 2)
  {
    if (R.p == 0 || R.p > N - 2)
      return NE;

    send[R.pred] = R.p - 1;
    send[1 - R.pred] = R.p + 1;

    if (receive[0] != R.p || receive[1] != R.p)
      return NE;
  }
else // D == 1
  {
    if (R.p == 0)
      send[0] = 1;
    else if (R.p == N - 1)
      send[0] = N - 2;
    else
      return NE;

    if (receive[0] != R.p)
      return NE;
  }

return ANO;
```

Pokud Hvězdné impérium tvoří kružnici, uvažme systém, kterému věštec vyvěští nejmenší hodnotu p. Tento systém jednomu ze svých sousedů pošle zprávu $p-1$, která je nutně menší než jemu vyvěštěná hodnota. Tento soused tedy nutně odpoví NE. Naopak, pokud Hvězdné impérium tvoří cestu a všichni věštci správně vyvěští pozici v posloupnosti, všichni odpoví ANO.

Dostáváme tedy řešení s $K = \lceil \log_2 N \rceil + 1$.