

## 58. ročník matematické olympiády na středních školách

---

### 21. mezinárodní olympiáda v informatice

In: Karel Horák (editor); Martin Mareš (editor); Peter Novotný (editor); Martin Panák (editor); Jaromír Šimša (editor); Jaroslav Švrček (editor); Pavel Töpfer (editor): 58. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 2008/2009. 50. mezinárodní matematická olympiáda. 21. mezinárodní olympiáda v informatice. (Czech). Praha: Jednota českých matematiků a fyziků, 2011. pp. 193–212.

Persistent URL: <http://dml.cz/dmlcz/405184>

### Terms of use:

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

## 21. mezinárodní olympiáda v informatice

Dvacátý první ročník Mezinárodní olympiády v informatice IOI 2009 se konal ve dnech 8.–15. 8. 2009 v bulharském Plovdivu. Záštitu nad soutěží převzal osobně bulharský prezident Georgi Parvanov, který se také zúčastnil slavnostního zahájení společně s ministryni školství, mládeže a vědy, paní Jordankou Fandakovou a s dalšími významnými hosty.

Na olympiádu do Bulharska přijely delegace z 82 zemí celého světa. Z každé země se olympiády mohou zúčastnit čtyři soutěžící a dva vedoucí, celkově letos soutěžilo 312 studentů. České družstvo bylo sestaveno na základě výsledků ústředního kola 58. ročníku Matematické olympiády — kategorie P a bylo tvořeno těmito studenty:

*Vlastimil Dort*, student gymnázia Špitálská v Praze,  
*Hynek Jemelík*, student gymnázia na tř. Kpt. Jaroše v Brně,  
*David Klačka*, student gymnázia na tř. Kpt. Jaroše v Brně,  
*Karel Tesař*, student VOŠ a SPŠE v Plzni.

Vedoucími české delegace byli doc. RNDr. *Pavel Töpfer*, CSc., a Mgr. *Zdeněk Dvořák*, Ph.D., oba z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze.

Již tradičně se naši účastníci připravovali na olympiádu společně s reprezentanty vybranými pro CEOI (Středoevropská olympiáda v informatice) na týdenním soustředěním CPSPC (Czech-Polish-Slovak Preparation Camp). Přípravné soustředění letos uspořádali polští organizátoři a zúčastnili se ho vybraní studenti ze všech tří zemí.

Během prvního dne pobytu v Bulharsku proběhlo slavnostní zahájení soutěže a studenti měli také příležitost seznámit se podrobně s počítači a se softwarovým prostředím, ve kterém budou pracovat při soutěži. Vlastní soutěž se konala jako obvykle ve dvou dnech, oddělených jedním dnem odpočinku. Po druhém soutěžním dnu následoval jeden den věnovaný výletu k moři a v závěrečném dni pobytu se uskutečnilo vyhlášení výsledků v historických prostorách antického amfiteátru.



Soutěž probíhá podobným způsobem jako praktická část ústředního kola naší Matematické olympiády — kategorie P. Každý soutěžící má přidělen osobní počítač, na kterém řeší zadané úlohy. V každém dni má na práci vymezen čas 5 hodin. Úlohy je třeba dovést až do tvaru odladěného programu, hotové programy se odevzdávají k vyhodnocení prostřednictvím soutěžního prostředí. Odevzdané programy se testují pomocí předem připravené sady testovacích dat. Prováděné testy jsou navíc omezeny časovými limity, aby se kromě otestování správnosti odlišila i časová efektivita algoritmu použitého jednotlivými účastníky soutěže. Při testování každé úlohy se používají sady testovacích dat různé velikosti, takže teoreticky zcela správné řešení založené na neefektivním algoritmu zvládne dokončit výpočet pouze pro některé, menší testy. Takové řešení je potom ohodnoceno částečným počtem bodů.

Úlohy bývají tradičně dost náročné a stávalo se, že řada účastníků nevyřešila správně ani jednu z nich. Novinkou letošního ročníku proto bylo přidání navíc jedné výrazně lehčí úlohy v každém soutěžním dnu. V každém dnu byly tedy zadány nikoliv tři jako dříve, ale čtyři úlohy. Tyto přidané snadnější úlohy byly vyhodnocovány ve zvláštním režimu „s plnou odezvou“ ihned v průběhu soutěže. Jedná se o podobný systém, jaký používáme od loňska u nás v kategorii P pro praktické úlohy domácího kola. Krátce po odevzdání vypracovaného programu do vyhodnocovacího systému se soutěžící dozví hodnocení svého řešení a má pak ještě možnost řešení opravit a odevzdat řešení téže úlohy opakovaně vícekrát. Soutěžící z programátorsky méně vyspělých zemí tak dostali velmi reálnou šanci vyřešit každý den aspoň některou úlohu, zatímco pro ty nejlepší nepředstavovala přidaná úloha žádné velké zdržení a mohli se pak věnovat ostatním, náročnějším problémům.

Každá z osmi soutěžních úloh byla hodnocena maximálně 100 body, takže celkově bylo možné získat až 800 bodů. To se ovšem nikomu nepodařilo, vítěz soutěže Genadii Karatzkevitch z Běloruska dosáhl výsledku 743 bodů. Na základě přesně stanovených pravidel se na IOI podle dosažených bodů rozdělují medaile. Některou z medailí obdrží nejvýše polovina účastníků soutěže, přičemž zlaté, stříbrné a bronzové medaile se udělují přibližně v poměru 1 : 2 : 3 (s ohledem na to, aby soutěžící se stejným bodovým ziskem získali stejnou medaili). Na letošní IOI bylo rozděleno celkem 26 zlatých, 50 stříbrných a 73 bronzových medailí. Výsledky našich soutěžících shrnuje následující tabulka.

Mezinárodní olympiáda v informatice je soutěží jednotlivců a žádné pořadí zúčastněných zemí v ní není vyhlášováno. Podle dosažených vý-

73. David Klaška	497 bodů	stříbrná medaile
94. Vlastimil Dort	458 bodů	bronzová medaile
140. Hynek Jemelík	403 bodů	bronzová medaile
260. Karel Tesař	179 bodů	–

sledků bychom se ale jistě umístili v lepší polovině. Nejúspěšnějšími zeměmi byly letos Čína a Korea se třemi zlatými medailemi.

Všechny podrobnosti o soutěži, texty soutěžních úloh i jejich řešení a celkové výsledky lze nalézt na adrese <http://www.ioi2009.org>. Příští ročníky IOI se budou konat v Kanadě (2010), Thajsku (2011), Itálii (2012) a Austrálii (2013). Pořadatelé IOI 2010 z Kanady na místě pozvali všechny delegace zúčastněné na IOI 2009, aby se zúčastnily také následujícího ročníku soutěže.

## Texty soutěžních úloh

### 1. Lukostřelba

Turnaj v lukostřelbě má následující pravidla. Do řady se postaví  $N$  terčů, očíslovaných zleva doprava od 1 do  $N$ . Soutěží  $2N$  lukostřelců. V každém kole turnaje se u každého terče utkají dva lukostřelci. Po skončení kola se lukostřelci přesunou tímto způsobem:

- ▷ Vítězi na terčích 2 až  $N$  včetně se posunou o jeden terč doleva (tj. k terčům 1 až  $N - 1$ ).
- ▷ Poražení na terčích 2 až  $N$  včetně a vítěz na terči 1 zůstávají u stejného terče.
- ▷ Poražený na terči 1 se přesune k terči  $N$ .

Turnaj trvá  $R$  kol. Počet kol je větší nebo roven počtu lučištníků (tj.  $R \geq 2N$ ).

Jste jediný účastník, který dorazil přesně na čas, nebo možná dokonce o pár minut pozdě (jako obvykle). Zbýlých  $2N - 1$  lučištníků zde už je a jsou seřazeni v pořadí, ve kterém nastoupí k terčům. Zbývá se již jen mezi ně někam zařadit, a turnaj může začít — dva nejlevější lučištníci nastoupí k terči 1, následující dva k terči 2, a tak dále, až dva nejpravější lučištníci nastoupí k terči  $N$ .

Všichni lučištníci včetně vás jsou oznámkováni dle dovednosti (jako ve škole, tj. lučištník s nižší známkou je dovednější). Žádní dva lučištníci nejsou stejně dovední, a když spolu soutěží, ten s nižší známkou vždy zvítězí.

Znáte známky všech soutěžících a chtěli byste se mezi ně zařadit tak, abyste skončili u terče s co nejmenším číslem. Je-li více možností, jak toho dosáhnout, vyberte tu, kde začínáte na terči s největším možným číslem.

*Úloha:* Napište program, který načte známky všech lučištníků včetně vás a počáteční pořadí vašich soupeřů a určí pozici, na niž byste se měli zařadit, splňující výše popsané podmínky.

*Omezení:*

- $1 \leq N \leq 200\,000$                       počet terčů, roven též polovině počtu lučištníků
- $2N \leq R \leq 1\,000\,000\,000$       počet kol turnaje
- $1 \leq S_k \leq 2N$                         známka lučištníku  $k$

*Vstup:* Váš program musí načíst ze standardního vstupu následující data:

- ▷ První řádek obsahuje dvě přirozená čísla  $N$  a  $R$ , oddělená mezerou.
- ▷ Následujících  $2N$  řádků obsahuje známky lučištníků. První z těchto řádků obsahuje vaši známku. Zbytek z nich obsahuje známky vašich soupeřů v pořadí, v němž jsou seřazeni (zleva doprava). Každý z těchto  $2N$  řádků obsahuje jedno přirozené číslo v rozsahu od 1 do  $2N$ . Zámka 1 je nejlepší a známka  $2N$  je nejhorší. Žádní dva lučištníci nemají stejnou známku.

*Výstup:* Váš program musí na standardní výstup vypsat jediný řádek obsahující jedno přirozené číslo v rozsahu od 1 do  $N$  včetně: číslo terče, na kterém budete v turnaji začínat.

*Hodnocení:* Pro některé ze vstupů bude  $N$  menší nebo rovno 5 000. Za tyto vstupy můžete získat až 60 bodů. Pro některé z těchto vstupů navíc bude  $N$  menší nebo rovno 200 a za tyto vstupy lze získat až 20 bodů.

*Příklady:*

<i>Vstup</i>	<i>Výstup</i>
4 8	3
7	
4	
2	
6	
5	
8	
1	
3	

Jste druhý nejhorší lukostřelec. Začnete-li na prvním terči, prohrajete a přesunete se na terč 4 a tam zůstanete až do konce. Podobně, začnete-li na terči 2 či 4, zůstanete na něm až do konce. Začnete-li na terči 3, porazíte nejhoršího lukostřelce, posunete se na terč 2 a tam zůstanete.

<i>Vstup</i>	<i>Výstup</i>
4 9	2
2	
1	
5	
8	
3	
4	
7	
6	

Jste druhý nejlepší lukostřelec. Nejlepší je již na prvním terči a zůstane tam až do konce. Kdekoliv začnete, budete se v každém kole posunovat o jeden terč doleva, až dorazíte k prvnímu terči. Zde prohrajete, přesunete se k čtvrtému terči, a opět se začnete přesunovat doleva. Abyste po 9 kolech byli na terči 1, musíte začít na terči 2.

## 2. Stavba

Potřebujete najmout dělníky na stavbu. Máte  $N$  zájemců, očíslovaných od 1 do  $N$ . V případě, že zaměstnáte zájemce číslo  $k$ , budete mu muset platit alespoň  $S_k$  korun. Navíc, zájemce číslo  $k$  má úroveň kvalifikace  $Q_k$ . Odbory vyžadují, aby platy byly přímo úměrné úrovni kvalifikace. Zaměstnáte-li například zájemce  $A$  a  $B$  takové, že  $Q_A = 3Q_B$ , pak budete muset dělníkovi  $A$  platit přesně třikrát tolik, co dělníkovi  $B$ . Platy mohou být i neceločíselné a dokonce je ani nemusí být možné vyjádřit v desítkové soustavě konečným počtem číslic; je tedy možné dělníkovi platit například třetinu či šestinu koruny.

Máte k dispozici  $W$  korun a chtěli byste najmout co nejvíc dělníků. Je jen na vás, koho najmete a kolik mu budete platit, ale musíte dodržet jak požadavky na minimální platy, tak omezení odborů. Všem vybraným dělníkům samozřejmě nemůžete dohromady platit více než  $W$  korun.

Vzhledem k tomu, že tito dělníci stejně budou jen nosit cihly, jejich kvalifikace pro vás není důležitá a snažíte se pouze maximalizovat počet najatých dělníků. Existuje-li více než jedna možnost, jak tohoto maximálního počtu dosáhnout, preferujete tu, v níž platíte dělníkům dohromady

nejméně peněz. Pokud existuje více možností, jak splnit tyto podmínky, zvolte jednu libovolnou z nich.

*Úloha:* Napište program, který načte požadavky na minimální mzdy a úroveň kvalifikace dělníků a částku, kterou disponujete, a určí, které dělníky máte najmout. Musíte najmout co nejvíce dělníků, a to tak, abyste přitom utratili co nejméně peněz a zároveň splnili výše popsané omezení odborů.

*Omezení:*

- $1 \leq N \leq 500\,000$  počet zájemců
- $1 \leq S_k \leq 20\,000$  minimální mzda požadovaná zájemcem číslo  $k$
- $1 \leq Q_k \leq 20\,000$  úroveň kvalifikace zájemce číslo  $k$
- $1 \leq W \leq 10\,000\,000\,000$  částka, již disponujete

*Důležité upozornění:* Největší možná hodnota  $W$  se nevejde do 32 bitů, musíte proto pro její reprezentaci použít 64-bitový datový typ, například `long long` v C/C++ či `int64` v Pascalu.

*Vstup:* Váš program musí načíst následující data ze standardního vstupu:

- ▷ První řádek obsahuje dvě přirozená čísla  $N$  a  $W$ , oddělená mezerou.
- ▷ Následujících  $N$  řádků popisuje zájemce. Na  $k$ -tém z těchto řádků se nachází dvě přirozená čísla  $S_k$  a  $Q_k$  oddělená mezerou.

*Výstup:* Váš program musí vypsát na standardní výstup následující data:

- ▷ První řádek musí obsahovat právě jedno přirozené číslo  $H$ , počet dělníků, které najmete.
- ▷ Následujících  $H$  řádků musí obsahovat čísla dělníků, které jste se rozhodli zaměstnat. Každý z těchto řádků musí obsahovat jedno přirozené číslo mezi 1 a  $N$ . Tato čísla musí být navzájem různá. Jejich pořadí může být libovolné.

*Hodnocení:* Za daný vstup obdržíte plný počet bodů, jestliže vaše volba dělníků splňuje všechna omezení, maximalizuje jejich počet a minimalizuje součet jejich platů. Jestliže váš výstup bude mít správný první řádek (tj. určíte správně hodnotu  $H$ ), ale zbytek nebude splňovat podmínky zadání, obdržíte 50 % bodů. Stejně budete hodnoceni i v případě, že zbytek výstupu nebude správně naformátován.

V několika z testovacích vstupů bude  $N$  nanejvýš 5 000. Celkově za tyto vstupy můžete získat až 50 bodů.

*Příklady:*

<i>Vstup</i>	<i>Výstup</i>
4 100	2
5 1000	2
10 100	3
8 10	
20 1	

Jediný způsob, jak si můžete dovolit najmout dva dělníky, je zaměstnat zájemce 2 a 3. Zájemci 2 zaplatíte 80 korun a zájemci 3 zaplatíte 8 korun. Celkově zaplacená částka bude menší než 100 korun, které máte k dispozici.

<i>Vstup</i>	<i>Výstup</i>
3 4	3
1 2	1
1 3	2
1 3	3

Zde si můžete dovolit zaměstnat všechny tři zájemce. Prvnímu z nich zaplatíte 1 korunu a zbylým dvěma 1 korunu a 50 haléřů. Dohromady jim tedy budete platit 4 koruny, což přesně odpovídá vašemu rozpočtu.

<i>Vstup</i>	<i>Výstup</i>
3 40	2
10 1	2
10 2	3
10 3	

Zde si nemůžete dovolit zaměstnat všechny tři zájemce, jelikož by vás to stálo 60 korun. Nicméně můžete najmout libovolné dva z nich. Vyberete si zájemce 2 a 3, jelikož na této kombinaci utratíte nejméně. Stačí vám dělníkovi 2 platit 10 korun a dělníkovi 3 platit 15 korun, dohromady tedy utratíte 25 korun. Oproti tomu, jestliže byste zaměstnali zájemce 1 a 2, museli byste jim platit 10 a 20 korun, a jestliže byste zaměstnali zájemce 1 a 3, platili byste jim 10 a 30 korun.

### 3. POI (úloha s úplnou odezvou)

Místní Plovdivská olympiáda v informatice (POI) má následující neobvyklá pravidla. Soutěží v ní  $N$  účastníků a mají za úkol řešit  $T$  soutěžních úloh. Každé odevzdané řešení úlohy je testováno s jedinými vstupními daty, takže za každou úlohu získá každý soutěžící buď plný počet bodů, nebo nic. Žádné částečné hodnocení neexistuje.

Počet bodů přidělených za správně vyřešenou úlohu se stanoví až po skončení soutěže a je roven počtu soutěžících, kteří tuto úlohu nevy-



řešili. Výsledné hodnocení soutěžícího je stanoveno jako součet bodů, které soutěžící získal za všechny jím vyřešené úlohy.

Filip se zúčastnil soutěže, ale je zmaten složitými pravidly bodování a nedokáže určit svoje umístění v celkovém pořadí POI. Pomozte Filipovi a napište mu program, který určí jeho celkové hodnocení a výsledné umístění v soutěži.

Před zahájením soutěže obdrželi účastníci jednoznačná identifikační čísla od 1 do  $N$ . Filip má číslo  $P$ . Ve výsledkové listině jsou soutěžící uspořádáni sestupně podle počtu získaných bodů. V případě shody bodů je ve výsledkové listině umístěn dříve ten soutěžící, který vyřešil více úloh. Pokud nerozhodne ani toto kritérium, soutěžící se stejnými výsledky budou uspořádáni vzestupně podle svých identifikačních čísel.

*Úloha:* Napište program, který určí Filipovo výsledné hodnocení a jeho pořadí ve výsledkové listině na základě informací, který soutěžící vyřešil kterou úlohu.

*Omezení:*

$1 \leq N \leq 2000$	počet účastníků soutěže
$1 \leq T \leq 2000$	počet řešených úloh
$1 \leq P \leq N$	Filipovo identifikační číslo

*Vstup:* Váš program musí přečíst ze standardního vstupu následující údaje:

- ▷ První řádek obsahuje tři celá čísla  $N$ ,  $T$ ,  $P$  oddělená vždy jednou mezerou.
- ▷ Dalších  $N$  řádků popisuje, které úlohy vyřešil který soutěžící. V pořadí  $k$ -tý z těchto řádků určuje, které úlohy vyřešil soutěžící s identifikačním číslem  $k$ . Každý takový řádek obsahuje  $T$  celých čísel oddělených mezerami. První z čísel na řádku udává, zda soutěžící  $k$  vyřešil první úlohu, druhé určuje, zda vyřešil druhou úlohu, atd. Každé z těchto  $T$  čísel je rovno 0 nebo 1, přičemž 1 znamená vyřešenou úlohu a 0 znamená nevyřešenou úlohu.

*Výstup:* Váš program musí zapsat na standardní výstup jediný řádek, který obsahuje dvě celá čísla oddělená jednou mezerou. První z nich představuje výsledný počet bodů, které Filip získal v soutěži. Druhé z čísel na výstupu určuje Filipovo pořadí ve výsledkové listině. Je to číslo z rozmezí od 1 do  $N$ , kde 1 znamená umístění na prvním místě (tj. soutěžící, který dosáhl nejvyššího bodového hodnocení) a  $N$  umístění na posledním místě (tj. soutěžící s nejnižším hodnocením).

*Hodnocení:* V testovacích datech odpovídajících celkovému ohodnocení 35 body nebude mít žádný soutěžící v POI stejný výsledný počet bodů jako Filip.

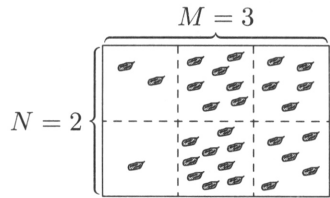
*Příklad:*

<i>Vstup</i>	<i>Výstup</i>
5 3 2	3 2
0 0 1	
1 1 0	
1 0 0	
1 1 0	
1 1 0	

První úlohu nevyřešil jediný soutěžící, takže bude hodnocena 1 bodem. Druhou úlohu nevyřešili dva soutěžící, bude tedy hodnocena 2 body. Třetí úlohu nevyřešili čtyři soutěžící, takže je hodnocena 4 body. První soutěžící tudíž získal celkem 4 body; druhý soutěžící (Filip), čtvrtý a pátý soutěžící mají všichni po 3 bodech; třetí soutěžící má 1 bod. Soutěžící s identifikačními čísly 2, 4 a 5 mají nejen shodný počet bodů, ale i stejný počet vyřešených úloh, takže jejich vzájemné pořadí bude určeno podle identifikačních čísel. Filip bude tudíž ve výsledkové listině POI na druhém místě, hned za soutěžícím s číslem 1.

#### 4. Rozinky

Cukrářka Bonny potřebuje rozřezat tabulku čokolády s rozinkami. Čokoláda má tvar obdélníku s dílky uspořádanými do  $N$  řádků a  $M$  sloupců. Celkově je tedy tvořena  $NM$  dílky. V každém dílku čokolády je jedna nebo více rozinek, žádná rozinka nepřesahuje hranice dílků.



Na začátku je čokoláda vcelku. Bonny ji potřebuje postupně řezat na menší a menší části, až nakonec bude čokoláda rozřezána až na jednotlivé samostatné dílky (tzn. na  $NM$  dílků). Bonny má hodně práce a s řezáním čokolády jí proto pomáhá její asistent Petr. Petr provádí pouze rovné řezy po hranicích dílků, vždy od kraje ke kraji řezaného kusu čokolády. Za každý provedený řez dostane Petr odměnu. Bonny nemá po ruce dostatek peněz, má ale spoustu zbývajících rozinek. Za každý provedený řez proto Bonny zaplatí Petrovi tolik rozinek, kolik rozinek je obsaženo v řezaném kusu čokolády.

Bonny chce zaplatit Petrovi za rozřezání celé čokolády na dílky co nejméně. Zná počet rozinek v každém z  $NM$  dílků a může podle toho

volit pořadí, v jakém dává Petrovi jednotlivé kusy čokolády na řezání, a určovat, jaký řez má Petr provést (zda vodorovně nebo svisle a kde přesně). Pomozte Bonny určit, jak má čokoládu nechat řezat, aby zaplatila Petrovi za její rozřezání na jednotlivé dílky co nejméně.

*Úloha:* Napište program, který na základě počtu rozinek obsažených v jednotlivých dílcích čokolády určí minimální počet rozinek, který musí Bonny zaplatit Petrovi za rozřezání celé čokolády na dílky.

*Omezení:*

$1 \leq N, M \leq 50$  rozměry čokolády (vyjádřeny počtem dílků)

$1 \leq R_{k,p} \leq 1\,000$  počet rozinek v dílku v  $k$ -té řadě a  $p$ -tém sloupci

*Vstup:* Váš program přečte ze standardního vstupu následující údaje:

- ▷ První řádek obsahuje dvě celá čísla  $N$  a  $M$  oddělená jednou mezerou.
- ▷ Dalších  $N$  řádků popisuje, kolik rozinek je obsaženo v jednotlivých dílcích čokolády. V pořadí  $k$ -tý z těchto  $N$  řádků popisuje  $k$ -tou řadu čokolády. Každý takový řádek obsahuje  $M$  celých čísel oddělených vždy jednou mezerou. Tato čísla určují počty rozinek obsažených v jednotlivých dílcích příslušné řady čokolády, a to v pořadí dílků zleva doprava.

*Výstup:* Váš program musí zapsat na standardní výstup jeden řádek obsahující jedno celé číslo: minimální počet rozinek, které musí Bonny zaplatit Petrovi.

*Hodnocení:* V několika z testovacích vstupů budou zadané hodnoty  $N$  a  $M$  nanejvýš rovny 7. Za tyto vstupy můžete získat celkem 25 bodů.

*Příklad:*

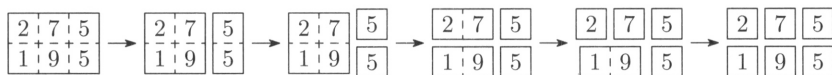
*Vstup*            *Výstup*

2 3            77

2 7 5

1 9 5

Jeden z více možných způsobů, jak lze dosáhnout ceny 77, vypadá následovně:



Bonny nejprve požádá Petra, aby oddělil třetí sloupec od zbytku čokolády. Za to zaplatí Petrovi 29 rozinek.

Potom dá Bonny Petrovi menší z obou kusů čokolády, který je tvořen dvěma dílky po 5 rozinkách. Petr ho rozdělí za 10 rozinek.

Dále dá Bonny Petrovi zbývající kus čokolády, který má dílky obsahující 2, 7, 1 a 9 rozinek. Bonny požádá Petra o vodorovný řez a zaplatí za něj 19 rozinek.

Konečně dá Bonny Petrovi na rozříznutí levý horní blok (řez za 9 rozinek) a pak i levý spodní blok (řez za 10 rozinek).

Celková cena řezání čokolády je  $29 + 10 + 19 + 9 + 10 = 77$  rozinek. Žádné jiné pořadí řezů nevede k levnějšímu rozřezání čokolády na jednotlivé dílky.

## 5. Garáž (úloha s úplnou odezvou)

Ve velké garáži je  $N$  parkovacích míst, která jsou očíslována od 1 do  $N$  včetně. Garáž otvírá každé ráno prázdná a je během dne provozována v následujícím režimu. Kdykoliv do garáže přijede auto, hlídač zkontroluje, zda je některé parkovací místo volné. Pokud není, auto musí počkat u vjezdu do garáže, dokud se nějaké místo neuvolní. Je-li v garáži více volných míst, auto zaparkuje na volném místě s nejmenším číslem. Jestliže do garáže přijedou další auta v době, kdy nějaké auto čeká na uvolnění místa, všechna tato auta se řadí u vjezdu do fronty v tom pořadí, v jakém přijela. Když se pak uvolní v garáži nějaké parkovací místo, zaparkuje na něm první auto z fronty (tzn. to z čekajících aut, které přijelo nejdříve).

Cena za parkování (v dolarech) se počítá jako součin váhy auta (v kilogramech) a koeficientu, jehož hodnota závisí na použitém parkovacím místě. Cena nezávisí na tom, jak dlouho auto zůstane v garáži.

Správce garáže ví, že dnes přijede do garáže celkem  $M$  aut, a zná také pořadí jejich příjezdů a odjezdů. Pomozte mu spočítat, kolik dolarů dnes celkově vybere za parkování.

*Úloha:* Napište program, který na základě znalosti cenových koeficientů jednotlivých parkovacích míst v garáži a údajů o příjíždějících autech (váhy aut, pořadí jejich příjezdů a odjezdů) určí celkovou cenu v dolarech, kterou řidiči parkujících aut zaplatí.

*Omezení:*

$1 \leq N \leq 100$       počet parkovacích míst

$1 \leq M \leq 2000$     počet aut

$1 \leq R_s \leq 100$     koeficient parkovacího místa  $s$  v dolarech za kilogram

$1 \leq W_k \leq 10000$  váha auta  $k$  v kilogramech

*Vstup:* Váš program musí přečíst ze standardního vstupu následující údaje:

▷ První řádek obsahuje celá čísla  $N$  a  $M$  oddělená mezerou.

- ▷ Dalších  $N$  řádků udává koeficienty parkovacích míst. V pořadí  $s$ -tý z těchto řádků obsahuje jedno celé číslo  $R_s$ , koeficient parkovacího místa číslo  $s$  v dolarech za kilogram.
- ▷ Následujících  $M$  řádků určuje váhy aut. Auta jsou očíslována od 1 do  $M$  včetně, a to bez ohledu na pořadí jejich příjezdu do garáže nebo odjezdu. V pořadí  $k$ -tý z těchto  $M$  řádků obsahuje jedno celé číslo  $W_k$ , váhu auta číslo  $k$  v kilogramech.
- ▷ Posledních  $2M$  řádků popisuje chronologicky, jak probíhají příjezdy a odjezdy jednotlivých aut. Kladné celé číslo  $i$  znamená, že auto číslo  $i$  přijíždí do garáže. Záporné celé číslo  $-i$  znamená, že auto číslo  $i$  odjíždí z garáže. Žádné auto nebude odjíždět z garáže dříve, než přijelo, a číslo každého auta od 1 do  $M$  včetně se objeví v této posloupnosti právě dvakrát — jednou při příjezdu a podruhé při odjezdu. Žádné auto neodjede z garáže před zaparkováním (tzn. žádné auto neopustí předčasně frontu, v níž čeká na parkování).

*Výstup:* Program musí zapsat na standardní výstup řádek obsahující jedno celé číslo — částku v dolarech, kterou dnes utrží správce garáže.

*Hodnocení:* V testovacích datech, za která lze získat hodnocení 40 bodů, bude mít každé přijíždějící auto k dispozici vždy alespoň jedno volné parkovací místo v garáži. V těchto případech tedy žádné auto nebude muset čekat na uvolnění místa.

*Příklady:*

<i>Vstup</i>	<i>Výstup</i>
2 4	16200
5	
2	
100	
500	
1000	
2000	
3	
1	
2	
4	
-1	
-3	
-2	
-4	

Auto číslo 3 zaparkuje na místě 1 a zaplatí  $1\,000 \cdot 5 = 5\,000$  dolarů.

Auto číslo 1 zaparkuje na místě 2 a zaplatí  $100 \cdot 2 = 200$  dolarů.

Auto číslo 2 přijede a musí čekat u vjezdu.

Auto číslo 4 přijede a musí čekat u vjezdu za autem číslo 2.

Když auto číslo 1 opustí své parkovací místo, auto číslo 2 na něm zaparkuje a zaplatí  $500 \cdot 2 = 1\,000$  dolarů.

Když auto číslo 3 opustí své parkovací místo, auto číslo 4 na něm zaparkuje a zaplatí  $2\,000 \cdot 5 = 10\,000$  dolarů.

*Vstup*      *Výstup*

3 4      5300

2

3

5

200

100

300

800

3

2

-3

1

4

-4

-2

-1

Auto číslo 3 zaparkuje na místě 1 a zaplatí  $300 \cdot 2 = 600$  dolarů.

Auto číslo 2 zaparkuje na místě 2 a zaplatí  $100 \cdot 3 = 300$  dolarů.

Auto číslo 1 zaparkuje na místě 1 (které se uvolnilo po odjezdu auta číslo 3) a zaplatí  $200 \cdot 2 = 400$  dolarů.

Auto číslo 4 zaparkuje na místě 3 (poslední zbývajícím místem) a zaplatí  $800 \cdot 5 = 4\,000$  dolarů.

## 6. Míša

Medvěd Míša objevil tajnou včelí zásobu medu a teď si lebedí a pochutnává si na něm. Vtom kde se vzala, tu se vzala — včela! Než se Míša stačil překulit a schovat za strom, včela ho zahlédla jedním ze svých očí a spustila poplach. Míša ví, že z úlů právě vyrazily hordy včel a pokud ho dopadnou, zle to s ním dopadne. Musí včas přestat jíst med a utéct do svého doupěte. Ale med je sladoučký a Míšovi se od něj vůbec nechce

odejít. Pomozte mu určit, jak dlouho ještě může zůstat, aby se stihl bezpečně vrátit domů.

Míšuň les vypadá jako čtvercová mřížka o rozměrech  $N$  krát  $N$ , orientovaná ve směru světových stran. Na každém políčku je strom, tráva, úl nebo Míšovo doupě. Dvě políčka *sousedí*, pokud se jedno z nich nachází bezprostředně na sever, jih, východ či západ od druhého (ale ne diagonálně). Jak jste si již jistě všimli, Míša je velmi mlsný a to se negativně projevuje na jeho pohybových schopnostech — každým krokem se dokáže posunout pouze na sousední políčko, a to pouze po trávě (na stromy Míša lézt nedokáže) či do svého doupěte. Samozřejmě se také nesmí posunout na políčko obsahující včely. Za jednu minutu Míša ujde nanejvýš  $S$  kroků.

V okamžiku, kdy včely vyhlásí poplach, je Míša na políčku obsahujícím med a včely jsou na všech políčkách, která obsahují úl (v lese může být několik úlů). Každou další minutu se odehrají následující události:

- ▷ Jestliže je Míša stále na políčku s medem, rozhodne se, zda zůstane další minutu nebo zda vyrazí domů. Jestliže zůstane, pak se celou minutu nepohne. Jinak (v případě, že se rozhodl vyrazit nebo již políčko s medem opustil) ujde nanejvýš  $S$  kroků dle výše popsanych pravidel.
- ▷ Poté, co Míša stráví minutu pojidáním medu či pohybem po lese, se včelí roj rozšíří o jedno pole do všech světových stran, ale jen po travnatých políčkách. Přesněji, roj včel se rozšíří na všechna travnatá políčka sousední s políčky, která již včely obsahují. Míša tedy nesmí být na žádném z těchto políček. Jakmile na nějakém políčku jsou včely, už na něm zůstanou až do konce — roj se nepřesunuje, ale zvětšuje se.

Na konci první minuty se tedy včely nacházejí na úlech a na travnatých políčkách, která s úly sousedí. Na konci druhé minuty jsou včely navíc na travnatých políčkách, na která se lze z úlů dostat dvěma kroky po travnatých políčkách ve směru světových stran, a tak dále. Po nějaké době včely zaplní všechna políčka, na která se z úlů dá dostat.

Ani Míša, ani včely nemohou opustit les. Dle výše popsanych pravidel Míša bude jíst med celočíselný počet minut.

*Úloha:* Napište program, který načte mapu lesa a určí maximální počet minut, které Míša může strávit na své výchozí pozici tak, aby se stále byl schopen vrátit do doupěte dřívě, než ho dostihnou včely.

*Omezení:*

$1 \leq N \leq 800$  velikost (délka strany) lesa

$1 \leq S \leq 1000$  maximální počet kroků, které Míša může urazit za 1 min

*Vstup:* Váš program musí ze standardního vstupu načíst následující data:

- ▷ První řádek obsahuje celá čísla  $N$  a  $S$  oddělená mezerou.
- ▷ Mapa lesa je obsažena na následujících  $N$  řádcích. Každý z těchto řádků obsahuje  $N$  znaků, z nichž každý popisuje jedno políčko lesa. Možné znaky a jejich význam:
  - T označuje strom
  - G označuje travnaté políčko
  - M označuje Míšovu počáteční pozici, která je na travnatém políčku
  - D označuje Míšovo doupě. Na toto políčko smí vstoupit Míša, ale ne včely.
  - H označuje úl

*Poznámka.* Mapa vždy bude obsahovat právě jedno písmeno M, právě jedno písmeno D a alespoň jedno písmeno H. Také vždy bude existovat posloupnost sousedících písmen G spojující počáteční pozici Míši s jeho doupětem a posloupnost sousedících písmen G spojující alespoň jeden úl s Míšovou počáteční pozicí. Tyto posloupnosti mohou být i prázdné, jestliže doupě či úl sousedí s Míšovou počáteční pozicí. Všimněte si, že včely nemohou vstoupit na políčko obsahující doupě, ani skrz něj nemohou projít. Toto políčko se tedy pro ně chová stejně jako strom.

*Výstup:* Váš program musí na standardní výstup vypsát jediný řádek obsahující jedno celé číslo: maximální počet minut, které Míša může strávit na své počáteční pozici tak, aby se stačil bezpečně vrátit domů.

Jestliže se Míša nemůže vrátit domů, aniž by ho včely dostihly, váš program musí místo toho vypsát na standardní výstup číslo  $-1$ .

*Hodnocení:* Za vstupy, pro něž je  $N$  nejvýše 60, lze získat až 40 bodů.

*Příklady:*

<i>Vstup</i>	<i>Výstup</i>
7 3	1
TTTTTTT	
TGGGGGT	
TGGGGGT	
MGGGGGD	
TGGGGGT	
TGGGGGT	
TGGGGGT	
TTHHHHT	



Poté, co Míša jí med jednu minutu, vyrazí nejkratší cestou přímo vpravo a po dvou minutách bezpečně dorazí domů.

Vzorový vstup      Vzorový výstup

7 3

2

TTTTTTT

TGGGGGT

TGGGGGT

MGGGGGD

TGGGGGT

TGGGGGT

TGHHGGT

Míša jí med dvě minuty, během třetí minuty provede kroky  $\rightarrow\uparrow\rightarrow$ , během čtvrté minuty kroky  $\rightarrow\rightarrow\rightarrow$  a nakonec kroky  $\downarrow\rightarrow$  během páté minuty.

## 7. Kraje

Výbor OSN pro rozvoj krajů (VOSNRK) má následující organizační strukturu: Zaměstnává celkem  $N$  lidí, očíslovaných od 1 do  $N$  v klesajícím pořadí dle služebního stáří. Služebně nejstarší zaměstnanec (předseda) má tedy číslo 1. Každý zaměstnanec kromě předsedy (zaměstnanec číslo 1) má právě jednoho přímého nadřízeného. Nadřízený je vždy služebně starší než všichni jeho podřízení. Zaměstnanec  $A$  je manažer zaměstnance  $B$ , jestliže  $A$  je buď přímý nadřízený zaměstnance  $B$ , nebo manažer přímého nadřízeného zaměstnance  $B$ . Předseda je tedy manažerem libovolného jiného zaměstnance. Zjevně také není možné, aby dva zaměstnanci byli navzájem svými manažery.

Každý zaměstnanec pochází z jednoho z  $R$  krajů. Kraje jsou očíslovány od 1 do  $R$ , přičemž jejich pořadí nemá žádný speciální význam. Úřad OSN pro vyšetřování (ÚOSNV) se zabývá podezřením, že VOSNRK diskriminuje některé kraje. Aby bylo možné toto podezření potvrdit či vyvrátit, ÚOSNV potřebuje počítačový systém schopný odpovídat na následující dotazy: Máte dány dva různé kraje  $r_1$  a  $r_2$ . Kolik existuje dvojic zaměstnanců  $e_1$  a  $e_2$  takových, že zaměstnanec  $e_1$  pochází z kraje  $r_1$ , zaměstnanec  $e_2$  pochází z kraje  $r_2$  a  $e_1$  je manažer zaměstnance  $e_2$ ? Každý dotaz má dva parametry (kraje  $r_1$  a  $r_2$ ) a odpověď na něj je jedno celé číslo (počet dvojic  $e_1$  a  $e_2$  splňujících výše popsané podmínky).

*Úloha:* Napište program, který načte informace o krajích, z nichž pochází zaměstnanci VOSNRK, a o přímých nadřízených těchto zaměstnanců a poté interaktivně odpovídá na dotazy výše popsaného typu.

*Omezení:*

- $1 \leq N \leq 200\,000$  počet zaměstnanců
- $1 \leq R \leq 25\,000$  počet krajů
- $1 \leq Q \leq 200\,000$  počet dotazů, na něž bude muset program odpovědět
- $1 \leq H_k \leq R$  kraj, ze kterého pochází zaměstnanec  $k$  ( $1 \leq k \leq N$ )
- $1 \leq S_k < k$  přímý nadřízený zaměstnance  $k$  ( $2 \leq k \leq N$ )
- $1 \leq r_1, r_2 \leq R$  čísla krajů, udávající parametry dotazu

*Vstup:* Váš program musí ze standardního vstupu načíst následující data:

- ▷ První řádek obsahuje celá čísla  $N$ ,  $R$  a  $Q$  oddělená jednotlivými mezerami.
- ▷ Následujících  $N$  řádků popisuje  $N$  zaměstnanců výboru v klesajícím pořadí dle služebního stáří, tj.  $k$ -tý z těchto řádků popisuje zaměstnance číslo  $k$ . První z řádků (popisující předsedu) obsahuje jedno celé číslo udávající kraj  $H_1$ , ze kterého předseda pochází. Každý z následujících  $N - 1$  řádků obsahuje dvě celá čísla oddělená mezerou: číslo  $S_k$  přímého nadřízeného zaměstnance  $k$  a číslo  $H_k$  kraje, z něhož zaměstnanec  $k$  pochází.

*Interakce:* Poté, co váš program načte výše popsaná vstupní data, musí střídavě číst dotazy ze standardního vstupu a vypisovat odpovědi na ně na standardní výstup. Na dotazy (jichž je dohromady  $Q$ ) je nutno odpovídat postupně — na každý dotaz musíte odpovědět předtím, než přečtete následující dotaz.

Každý dotaz je zadán na jednom řádku, obsahujícím dvě navzájem různá celá čísla  $r_1$  a  $r_2$  oddělená mezerou. Odpověď na tento dotaz musí být vypsána na standardní výstup jako jeden řádek obsahující jedno celé číslo: počet dvojic zaměstnanců  $e_1$  a  $e_2$  takových, že zaměstnanec  $e_1$  pochází z kraje  $r_1$ , zaměstnanec  $e_2$  pochází z kraje  $r_2$  a  $e_1$  je manažer zaměstnance  $e_2$ .

*Poznámka:* Vstupní data budou taková, že správnou odpovědí na každý zadaný dotaz bude číslo menší než 1 000 000 000.

*Důležitá poznámka:* Aby váš program správně komunikoval s vyhodnocovačem, je nutné, aby „spláchnul“ (flush) standardní výstup po každé odpovědi. Musí také zabránit zablokování při čtení standardního vstupu, což by mohlo nastat, pokud byste použili například příkaz `scanf ("%d\n")`.

*Hodnocení:* Za vstupy, pro něž  $R$  nepřesáhne 500, lze získat až 30 bodů.

Za vstupy, pro něž z každého kraje bude pocházet nanejvýš 500 zaměstnanců, lze získat až 55 bodů.

Za vstupy splňující obě podmínky lze získat 15 bodů a za vstupy, které splňují alespoň jednu z těchto podmínek, lze získat 70 bodů.

*Příklad:*

Vstup      Výstup

6 3 4

1

1 2

1 3

2 3

2 3

5 1

1 2

1 [spláchnout standardní výstup]

1 3

3 [spláchnout]

2 3

2 [spláchnout]

3 1

1 [spláchnout]

*Testování:* Jestliže budete testovat své řešení pomocí rozhraní soutěžního prostředí, vstupní soubor, který mu poskytnete, musí obsahovat vstupní data i dotazy (obdobně jako ve výše uvedeném příkladu, ale bez prázdných řádků).

## 8. Obchodní cestující

Obchodní cestující usoudil, že optimální plánování cesty mezi městy po zemi je výpočetně příliš náročné, a proto přenesl svůj obchod na řeku Dunaj. Při plánování svých obchodních cest se tak může omezit pouze na jednorozměrný případ, neboť se pohybuje výhradně podél vodního toku. Pořídil si velmi rychlý člun, s jehož pomocí se dostane po řece odkudkoliv kamkoliv v zanedbatelném čase; člun má ovšem značnou spotřebu paliva. Na každý metr ujetý proti proudu (tzn. směrem k prameni řeky) spotřebuje palivo v ceně  $U$  dolarů, zatímco na každý metr ujetý po proudu (tzn. směrem od pramene) spotřebuje palivo za  $D$  dolarů.

Obchodní cestující by rád navštívil  $N$  trhů, které se konají na různých místech podél řeky. Každý trh se koná pouze jeden den. O každém trhu  $X$  zná obchodní cestující jeho datum  $T_X$  (je vyjádřeno počtem dnů od doby,

kdy si pořídil člun), místo konání  $L_X$  (je vyjádřeno jako vzdálenost v metrech podél toku od pramene řeky) a předpokládanou tržbu v dolarech  $M_X$ , kterou obchodní cestující získá, pokud tento trh navštíví. Obchodní cestu musí zahájit i zakončit ve svém domovském přístavu na řece, který se nachází v místě  $S$  (měřeno rovněž v metrech podél toku od pramene řeky).

Pomozte obchodnímu cestujícímu zvolit, které trhy má navštívit (pokud vůbec nějaké) a v jakém pořadí, aby svou obchodní cestou dosáhl co nejvyššího zisku. Zisk obchodního cestujícího je roven součtu tržeb (v dolarech), které získá na všech navštívených trzích, minus celková cena paliva (rovněž v dolarech), které spotřebuje jeho člun při cestování po řece.

Pamatujte na to, že pokud se trh  $A$  koná dříve než trh  $B$ , obchodní cestující je může navštívit jedině v tomto pořadí (tzn. nemůže navštívit trh  $B$  a teprve potom trh  $A$ ). Jestliže se dva trhy konají ve stejném dni, obchodní cestující může navštívit oba tyto trhy, a to v libovolném pořadí. Není nijak omezeno, kolik trhů může obchodní cestující navštívit v jednom dnu, ale přirozeně nemůže navštívit dvakrát stejný trh a získat na něm dvakrát tržbu. Může ale projet místem, kde se právě koná trh, na kterém již dříve byl (bez získání další tržby na tomto trhu).

*Úloha:* Napište program, který určí maximální zisk, jehož může dosáhnout obchodní cestující na své cestě. Vstupem programu jsou údaje o datu, místu konání a předpokládané tržbě pro všechny probíhající trhy, o umístění domovského přístavu a o jízdních nákladech člunu.

*Omezení:*

$1 \leq N \leq 500\,000$  počet trhů

$1 \leq D \leq U \leq 10$  cena paliva (v dolarech) za jeden metr jízdy  
proti proudu ( $U$ ) a po proudu ( $D$ )

$1 \leq S \leq 500\,001$  umístění domu obchodního cestujícího

$1 \leq T_k \leq 500\,000$  datum konání trhu  $k$

$1 \leq L_k \leq 500\,001$  místo konání trhu  $k$

$1 \leq M_k \leq 4\,000$  předpokládaná tržba (v dolarech) při návštěvě trhu  $k$

*Vstup:* Váš program musí přečíst ze standardního vstupu následující údaje:

- ▷ První řádek obsahuje celá čísla  $N$ ,  $U$ ,  $D$  a  $S$  v uvedeném pořadí, oddělená vždy jednou mezerou.
- ▷ Dalších  $N$  řádků popisuje všech  $N$  trhů (v libovolném vzájemném uspořádání). V pořadí  $k$ -tý z těchto  $N$  řádků popisuje trh číslo  $k$ ;

obsahuje tři celá čísla oddělená vždy jednou mezerou: datum konání trhu  $T_k$ , jeho umístění  $L_k$  a předpokládanou tržbu obchodního cestujícího  $M_k$ .

*Poznámka:* Všechna umístění uvedená na vstupu jsou navzájem různá. To znamená, že žádné dva trhy se nekonají na stejném místě a žádný trh se nekoná v místě, kde má obchodní cestující svůj domovský přístav.

*Výstup:* Váš program musí zapsat na standardní výstup jeden řádek obsahující jedno celé číslo: maximální zisk, jakého může obchodní cestující dosáhnout uskutečněním obchodní cesty.

*Hodnocení:* V testovacích datech ohodnocených 60 body se žádné dva trhy nebudou konat ve stejném dni.

V testovacích datech ohodnocených 40 body nebude žádné číslo na vstupu větší než 5 000.

Obě výše uvedené podmínky budou zároveň splněny v testovacích datech ohodnocených 15 body.

Alespoň jedna z obou výše uvedených podmínek bude splněna v testovacích datech ohodnocených 85 body.

*Příklad:*

<i>Vstup</i>	<i>Výstup</i>
4 5 3 100	50
2 80 100	
20 125 130	
10 75 150	
5 120 110	

Optimální je navštívit trhy 1 a 3 (tzn. trhy na pozicích 80 a 75).

Pořadí událostí a jejich finanční důsledky budou následující:

- ▷ Obchodní cestující pojedje 20 metrů proti proudu za 100 dolarů. Dosavadní zisk:  $-100$
- ▷ Navštíví trh číslo 1, kde utrží 100. Dosavadní zisk: 0
- ▷ Pojedje 5 metrů proti proudu za 25. Dosavadní zisk:  $-25$
- ▷ Navštíví trh číslo 3, kde utrží 150. Dosavadní zisk: 125
- ▷ Pojedje 25 metrů po proudu, aby se vrátil domů, což stojí 75. Výsledný zisk: 50

