17. mezinárodní olympiáda v informatice

Nowy Sącz, Poland 2005
The 17-th International Olympiad in Informatics

V pořadí 17. ročník mezinárodní olympiády v informatice (IOI — International Olympiad in Informatics) se konal v Polsku ve dnech 18.–25. 8. 2005. Místem konání bylo město Nowy Sącz, které se nachází na samém jihu Polska nedaleko slovenských hranic. Celá soutěž probíhala v prostorách první polské soukromé vysoké školy WSB-NLU (Wyższa Szkola Biznesu — National Louis University), v blízkých kolejích školy bylo zajištěno i ubytování všech účastníků. Organizátoři ovšem byli většinou pracovníky varšavské university.

Letošní olympiády se zúčastnilo 276 soutěžících ze 72 zemí celého světa, další čtyři země vyslaly své pozorovatele s cílem seznámit se s průběhem soutěže a zúčastnit se jí aktivně v příštím roce. Každé družstvo je tvořeno čtyřmi soutěžícími studenty a je doprovázeno dvěma pedagogickými pracovníky jako vedoucími. Českou republiku letos reprezentovalo družstvo ve složení:

Ondřej Bílka, student Gymnázia ve Zlíně, Lesní čtvrť,
Jan Bulánek, absolvent Gymnázia J. Vrchlického v Klatovech,
Zbyněk Falt, absolvent Gymnázia ve Žďáru nad Sázavou,
Daniel Marek, student Gymnázia Ch. Dopplera v Praze 5.

Naši soutěžící byli vybráni na základě svých výsledků dosažených v celostátním kole kategorie P 54. ročníku matematické olympiády. Vedoucím české delegace byl jmenován doc. RNDr. *Pavel Töpfer*, CSc., z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze, druhým vedoucím byl Mgr. *Zdeněk Dvořák*, rovněž z MFF UK v Praze.

Vlastní soutěž IOI probíhá vždy ve dvou soutěžních dnech, v každém z nich soutěžící řeší po dobu pěti hodin tři zadané úlohy. Každý účastník má pro svoji práci přidělen osobní počítač s nainstalovanými překladači programovacích jazyků Pascal, C a C++ a s interaktivním webovým rozhraním pro komunikaci soutěžícího s řídicím a vyhodnocovacím systémem soutěže. To umožňuje zálohovat data, tisknout výpisy programů, ověřovat správnost chování programu a zejména pak předávat vytvořené programy na vyhodnocení. Všechny soutěžní úlohy jsou algoritmického charakteru a je nutné dovést je až do podoby kompletního odladěného programu, podobně jako je tomu třeba i v praktické části celostátního kola kategorie P naší matematické olympiády.

Odevzdané programy jsou vždy po skončení soutěžního dne automaticky testovány pomocí předem připravené sady testovacích dat, aby se ověřila jejich správnost. Důležitou součástí těchto testů jsou časové limity. Je pevně stanoveno, jak nejdéle může program počítat pro každá vstupní data. Tímto způsobem se mezi správně fungujícími programy ještě rozliší, na jak dobrém algoritmu je který program založen. Některá vstupní data zadávaná při testování jsou malá, takže výpočet s nimi stihne v časovém limitu i pomalejší algoritmus, naopak jiná vstupní data jsou rozsáhlá a včas je zvládne zpracovat jedině program využívající dostatečně efektivní algoritmus.

Vedle samotné soutěže byl pro všechny účastníky IOI 2005 připraven i zajímavý doprovodný program. Jeho součástí byl celodenní výlet do Krakowa a do známých solných dolů v městě Wieliczka, druhý výlet mířil do Národního parku Pieniny a zahrnoval i dvouhodinovou plavbu na pltích po řece Dunajec.

Letošní soutěžní úlohy byly algoritmicky zajímavé, byly dobře zvolené a pečlivě připravené. Nalézt opravdu dobré efektivní řešení bylo většinou poměrně obtížné, většina úloh však umožňovala získat částečné bodové ohodnocení i za méně efektivní algoritmus, jehož návrh tak náročný nebyl. Bodová ohodnocení získaná v soutěži se tak velmi dobře rozložila téměř v celé škále od nuly do plného počtu bodů. Za každou úlohu bylo možné získat maximálně 100 bodů, tj. celkově v soutěži 600 bodů. Hned čtyřem studentům (dva z Číny, po jednom z USA a Ukrajiny) se podařilo vyřešit všech šest soutěžních úloh zcela bezchybně a získali plný počet 600 bodů.

Na základě dosažených výsledků se na IOI udělují mediale tak, že nejvýše polovina účastníků obdrží některou z medailí, přičemž počet zlatých, stříbrných a bronzových medialí je v rámci možností přibližně v poměru 1 : 2 : 3. Letos bylo uděleno 24 zlatých medailí (soutěžícím, kteří dosáhli alespoň 496 bodů), 47 stříbrných medailí (za zisk alespoň 393 bodů) a 67 bronzových medailí (pro ty, kdo v soutěži získali minimálně 275 bodů). O medaile se podělili zástupci 57 zemí ze 72 zúčastněných. Našim studentům se tentokrát vedlo o něco hůře než v minulých letech, v soutěži získali jen dvě bronzové medaile.

Výsledky našich studentů:

82.	Daniel Marek	375 bodů	bronzová medaile
112.	Ondřej Bílka	316 bodů	bronzová medaile
	Jan Bulánek	233 bodů	–
	Zbyněk Falt	203 bodů	–

Mezinárodní olympiáda v informatice je soutěží jednotlivců, takže žádné oficiální pořadí národních družstev se v ní nevyhlašuje. Není ani stanoveno, zda by se mělo určovat podle součtu dosažených bodů, součtu umístění nebo třeba podle počtu získaných medailí. Podle kterékoliv zvolené metodiky výpočtu by se Česká republika umístila kolem 30.–35. místa, tzn. někde ke konci lepší poloviny zúčastněných zemí. Nejúspěšnějšími zeměmi letošního ročníku IOI byly po řadě Čína, USA a Slovensko, jejichž všichni čtyři soutěžící získali zlaté medaile. Další místa v pořadí první desítky by obsadily Thajsko, Ukrajina, Korea, Rusko, Polsko, Kanada a Izrael.

Příští, osmnáctý ročník soutěže IOI se bude konat v druhé polovině srpna 2006 v Mexiku. Uskuteční se ve městě Mérida v naprosto stejném místě, kde úspěšně proběhla 46. mezinárodní matematická olympiáda, o níž se dočtete na předchozích stránkách.

Texty soutěžních úloh

1. Hory

V Horském zábavním parku si pořídili zbrusu novou horskou dráhu. Její trať se skládá z n kolejnic spojených na koncích. Začátek první kolejnice je upevněn ve výšce 0. Bajtazar, operátor horské dráhy, může změnit sklon libovolného souvislého úseku kolejnic a tak upravit trasu horské dráhy. Sklon ostatních kolejnic se přitom nemění, takže se musí změnit výška, v níž se nacházejí kolejnice následující po úseku, jehož sklon Bajtazar upravil. Tak se zajistí, že se trať nepřeruší a její začátek zůstane ve výšce 0. Na obrázcích na druhé straně vidíte dva příklady úpravy horské dráhy.

Na trať vyjíždí vůz s energií dostatečnou na vystoupání do výšky h. Vůz tedy pokračuje v jízdě tak dlouho, dokud jeho výška nepřesáhne h nebo dokud nedojede na konec dráhy. Je zadán seznam všech jízd a úprav dráhy v průběhu jednoho dne. Určete pro každou jízdu, kolik kolejí vůz projel, než se zastavil.

Stav horské dráhy je reprezentován posloupností $d_1, \ldots, d_n$ změn výšky. Číslo d_i představuje rozdíl výšek začátku a konce i-té kolejnice v centimetrech. Pokud je vůz na konci $(i-1)$-ní kolejnice ve výšce h centimetrů, pak po projetí i-té kolejnice je ve výšce $h + d_i$ centimetrů.

Na začátku dne je horská dráha rovná, a tedy $d_i = 0$ pro všechna i. Během dne probíhají v libovolném pořadí úpravy dráhy a jízdy. Každá úprava dráhy je popsána jako trojice celých čísel a, b a D. Po provedení této úpravy je $d_i = D$ pro všechna i taková, že $a \leqq i \leqq b$, zatímco sklon ostatních kolejnic (tedy d_i pro i mimo tento interval) se nemění.

Každá jízda je popsána jedním celým číslem h — největší výškou, které může vůz dosáhnout.

Úloha:

Napište program, který:
- ▷ Načte ze standardního vstupu posloupnost úprav dráhy a jízd.
- ▷ Pro každou jízdu určí na základě aktuální konfigurace trati počet projetých kolejnic.
- ▷ Vypíše odpovědi na standardní výstup.

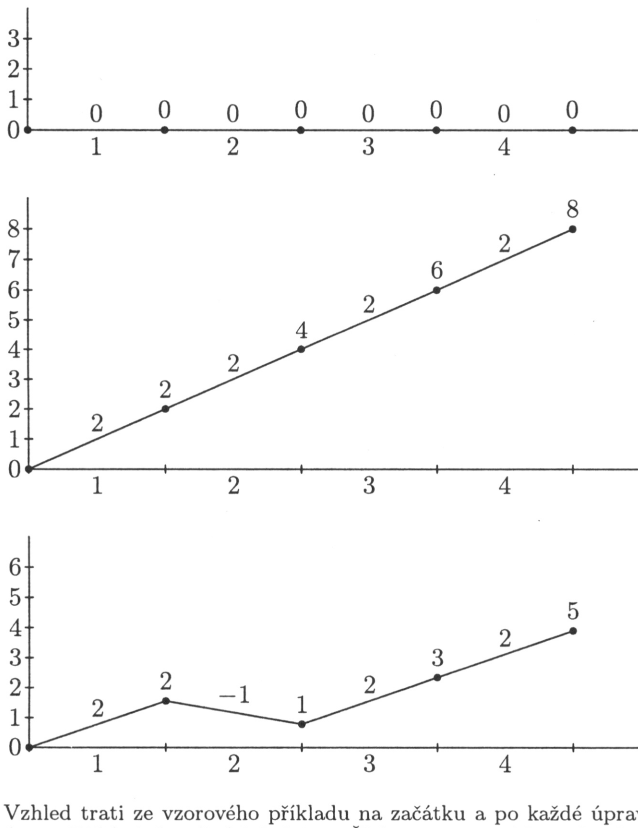
Vstup: První řádek vstupu obsahuje jedno kladné celé číslo n — počet kolejnic, $1 \leqq n \leqq 1\,000\,000\,000$. Následující řádky obsahují úpravy dráhy a jízdy v pořadí, ve kterém nastaly během dne. Na posledním řádku vstupu je ukončovací značka. Každý řádek obsahuje jednu z těchto možností:
- ▷ Úprava dráhy — písmeno `I`, a trojici celých čísel a, b, D, vše odděleno vždy jednou mezerou ($1 \leqq a \leqq b \leqq n$, $-1\,000\,000\,000 \leqq D \leqq 1\,000\,000\,000$).
- ▷ Jízda — písmeno `Q`, a jedno celé číslo h ($0 \leqq h \leqq 1\,000\,000\,000$) oddělené jednou mezerou.
- ▷ Koncová značka — písmeno `E`, označuje konec vstupních dat.

Výška všech částí dráhy je v libovolném okamžiku v intervalu $[0, 1\,000\,000\,000]$ centimetrů. Vstup obsahuje nejvýše 100 000 řádků. Pro 50 % vstupních dat platí, že $1 \leqq n \leqq 20\,000$ a že vstup má nejvýše 1 000 řádků.

Výstup: V pořadí i-tý řádek výstupu obsahuje jedno celé číslo — počet kolejnic, které vůz projel při i-té jízdě.

Příklad vstupu a výstupu:

```
4                         4
Q 1                       1
I 1 4 2                   0
Q 3                       3
Q 1
I 2 2 -1
Q 3
E
```



Obr. 62. Vzhled trati ze vzorového příkladu na začátku a po každé úpravě. Čísla na ose x představují čísla jednotlivých kolejnic. Čísla na ose y a čísla nad tečkami označují výšku. Čísla nad úsečkami označují změny výšky.

2. Posloupnost

Mějme neklesající posloupnost celých čísel $s_1, \ldots, s_{n+1}$ ($s_i \leqq s_{i+1}$ pro $1 \leqq i \leqq n$). Posloupnost $m_1, \ldots, m_n$ definovanou pro $1 \leqq i \leqq n$ předpisem $m_i = \frac{1}{2}(s_i + s_{i+1})$ nazveme *průměrovou posloupností* pro posloupnost $s_1, \ldots, s_{n+1}$. Například posloupnost 1,5, 2, 3 je průměrovou posloupností pro posloupnost 1, 2, 2, 4. Členy průměrové posloupnosti nemusí být obecně celá čísla. V této úloze se však budeme zabývat jen průměrovými posloupnostmi tvořenými pouze celými čísly.

Je zadána neklesající posloupnost n celých čísel $m_1, \ldots, m_n$. Určete počet různých neklesajících posloupností $s_1, \ldots, s_{n+1}$ tvořených $n+1$ celými čísly tak, aby daná posloupnost $m_1, \ldots, m_n$ byla jejich průměrovou posloupností.

Úloha:

Napište program, který:

- ▷ Načte ze standardního vstupu neklesající posloupnost $m_1, \ldots, m_n$ celých čísel.
- ▷ Určí počet neklesajících posloupností, pro které je $m_1, \ldots, m_n$ průměrovou posloupností.
- ▷ Vypíše tento počet na standardní výstup.

Vstup: První řádek vstupu obsahuje obsahuje jedno celé číslo n ($2 \leqq n \leqq 5\,000\,000$). Následujících n řádků obsahuje posloupnost $m_1, \ldots, m_n$. Hodnota m_i ($0 \leqq m_i \leqq 1\,000\,000\,000$) je zadána na řádku číslo $i + 1$. Alespoň pro 50 % vstupních dat navíc platí $n \leqq 1\,000$ a $m_i \leqq 20\,000$.

Výstup: Program vypíše na standardní výstup právě jedno celé číslo — počet neklesajících celočíselných posloupností, pro něž je posloupnost zadaná na vstupu posloupností průměrovou.

Příklad vstupu a výstupu:

3 2 5 9	4

Existují totiž právě čtyři neklesající celočíselné posloupnosti, pro které je 2, 5, 9 průměrovou posloupností. Tyto posloupnosti jsou:

▷ 2, 2, 8, 10	1, 3, 7, 11
▷ 0, 4, 6, 12	−1, 5, 5, 13

3. Zahrada

Bajtazar má nejkrásnější zahradu v Bajticích. Vysadil si tam n růží, které mu v létě nádherně rozkvetly. Bajtazar začal podnikat v zábavním průmyslu, a proto už nemá dost času na to, aby se staral o všechny růže. Chtěl by si najmout dva zahradníky, kteří by mu pomáhali. Každý z těchto zahradníků by se staral o jednu obdélníkovou část zahrady. Aby se zahradníci nehádali, obě části musí být disjunktní a každá z nich musí obsahovat právě k růží.

Bajtazar také chce tyto obdélníkové oblasti oplotit, ale má málo peněz, takže chce, aby ploty byly co nejkratší. Vaším úkolem je pomoci Bajtazarovi vybrat nejvhodnější umístění těchto oblastí.

Zahrada má tvar obdélníka l metrů dlouhého a w metrů širokého. Je rozdělena na $l \cdot w$ čtverců o velikosti 1×1 metr. Zvolíme si systém souřadnic s osami rovnoběžnými se stranami zahrady. Všechny čtverce jsou označeny celočíselnými souřadnicemi (x, y) takovými, že $1 \leqq x \leqq l$ a $1 \leqq y \leqq w$. Každý čtverec může obsahovat libovolný počet růží.

Hledané obdélníkové oblasti musí mít strany rovnoběžné se stranami zahrady a čtverce v jejich rozích musí mít celočíselné souřadnice. Pro $1 \leqq l_1 \leqq l_2 \leqq l$ a $1 \leqq w_1 \leqq w_2 \leqq w$ obdélníková oblast s rohy (l_1, w_1), (l_1, w_2), (l_2, w_1) a (l_2, w_2):

- ▷ obsahuje čtverce se souřadnicemi (x, y) takovými, že $l_1 \leqq x \leqq l_2$ a $w_1 \leqq y \leqq w_2$, a
- ▷ má obvod $2(l_2 - l_1 + 1) + 2(w_2 - w_1 + 1)$.

Obě obdélníkové oblasti musí být navzájem disjunktní, tzn. nemohou obsahovat tentýž čtverec. Mohou mít společný okraj nebo jeho část, ale i v takovém případě musí být každá oblast ohraničena vlastním plotem.

Úloha:

Napište program, který:

- ▷ Načte ze standardního vstupu rozměry zahrady, počet růží a jejich umístění v zahradě, a počet růží, které musí být v každé obdélníkové oblasti.
- ▷ Určí dvě obdélníkové oblasti tak, aby součet jejich obvodů byl co nejmenší, a zároveň aby splňovaly ostatní podmínky.
- ▷ Vypíše na standardní výstup součet jejich obvodů, nebo slovo NO, pokud takové dvě oblasti neexistují.

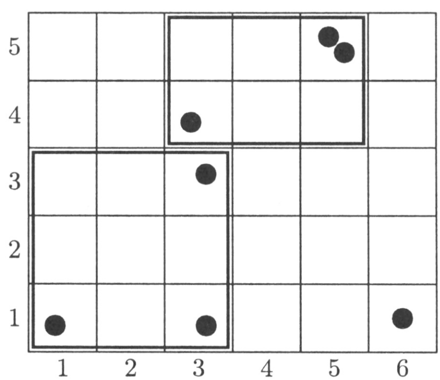
Vstup: První řádek vstupu obsahuje obsahuje dvě celá čísla l a w $(1 \leqq l,\ w \leqq 250)$ oddělená jednou mezerou — délku a šířku zahrady. Druhý řádek obsahuje dvě celá čísla n a k $(2 \leqq n \leqq 5\,000,\ 1 \leqq k \leqq \frac{1}{2}n)$

oddělená jednou mezerou — počet růží v zahradě a počet růží, které by měly být v každé obdélníkové oblasti. Následujících n řádků obsahuje souřadnice růží, na každém z nich je uvedeno umístění jedné růže. Řádek $(i+2)$-hý obsahuje dvě celá čísla l_i, w_i ($1 \leqq l_i \leqq l$, $1 \leqq w_i \leqq w$) oddělená jednou mezerou — souřadnice čtverce obsahujícího i-tou růži. V každém čtverci se může nacházet více růží. Pro 50 % vstupních dat platí $l, w \leqq 40$.

Výstup: Program vypíše na standardní výstup právě jedno celé číslo — nejmenší možný součet obvodů dvou disjunktních obdélníkových oblastí, z nichž každá obsahuje právě k růží, nebo slovo NO, pokud takové oblasti neexistují.

Příklad vstupu a výstupu:

```
6 5                    22
7 3
3 4
3 3
6 1
1 1
5 5
5 5
3 1
```



Obr. 63.

4. Narozeniny

Bajtazar má dnes narozeniny. Na oslavě jeho narozenin se sešlo n dětí (včetně Bajtazara). Děti jsou označeny čísly od 1 do n. Bajtazarovi rodiče připravili velký kulatý stůl a kolem něj rozmístili n židlí. Každé dítě se po příchodu hned posadí k tomuto stolu. Dítě číslo 1 si sedne na jednu z židlí. Dítě číslo dvě se posadí hned vedle dítěte číslo 1, po jeho levé ruce. Dítě číslo 3 si sedne nalevo od dítěte číslo 2, atd., až nakonec dítě číslo n se posadí na poslední volnou židli mezi dětmi číslo 1 a $n-1$.

Bajtazarovi rodiče vědí, že některé děti zlobí, pokud sedí příliš blízko u sebe. Proto chtějí přesadit děti v jiném pořadí. Toto pořadí je popsáno permutací $p_1, p_2, \ldots, p_n$ ($p_1, p_2, \ldots, p_n$ jsou navzájem různá celá čísla od 1 do n) — dítě číslo p_1 má sedět mezi dětmi číslo p_n a p_2, dítě p_i (pro $i = 2, 3, \ldots, n-1$) má sedět mezi dětmi číslo p_{i-1} a p_{i+1}, a dítě číslo p_n má sedět mezi dětmi číslo p_{n-1} a p_1. Všimněte si, že dítě číslo p_1 může sedět jak nalevo, tak napravo od dítěte p_n.

Všechny děti už ale dorazily a posadily se ke stolu. Aby děti seděly podle představ Bajtazarových rodičů, musí se každé dítě přesunout o nějaký počet míst doleva či doprava. Rodiče musí pro každé dítě určit, kam se má přemístit - tj. musí určit směr pohybu (doleva nebo doprava) a vzdálenost (počet míst). Na smluvený signál všechny děti vstanou, dojdou na určené místo a znovu si sednou.

Každý si asi uvědomil, že po vydání signálu nastane naprostý chaos. Míra chaosu je určena největší vzdáleností, o jakou se musí některé z dětí posunout. Děti mohou být přesazeny různými způsoby. Rodiče se rozhodli vybrat z nich ten, jehož míra chaosu je co nejmenší. Pomozte jim nalézt takový způsob přesazení dětí.

Úloha:

Napište program, který:

▷ načte ze standardního vstupu počet dětí a permutaci popisující požadované pořadí dětí u stolu,

▷ určí nejmenší možné m takové, že existuje způsob přesazení dětí, při kterém se každé dítě posune nejvýše o m míst,

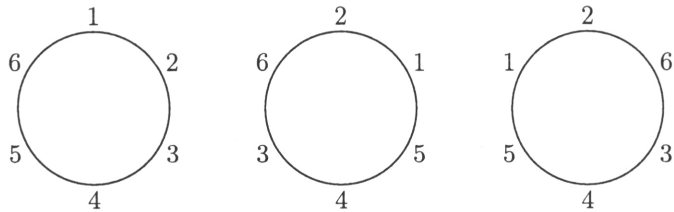
▷ vypíše výsledek na standardní výstup.

Vstup: První řádek standardního vstupu obsahuje jedno celé číslo n ($1 \leqq n \leqq 1\,000\,000$). Druhý řádek obsahuje n celých čísel $p_1, p_2, \ldots, p_n$ oddělených vždy jednou mezerou. Čísla $p_1, p_2, \ldots, p_n$ tvoří permutaci množiny $\{1, 2, \ldots, n\}$, která popisuje požadované pořadí dětí u stolu. Pro 50 % vstupních dat navíc platí, že $n \leqq 1\,000$.

Výstup: Program vypíše na standardní výstup právě jedno celé číslo — nejmenší možnou míru chaosu m.

Příklad vstupu a výstupu:

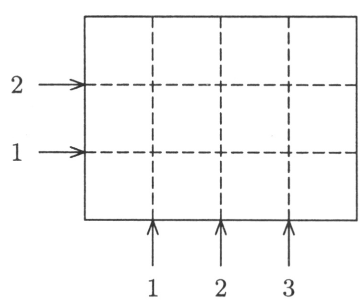
```
6                  2
3 4 5 1 2 6
```



Obr. 64. Na levém obrázku je znázorněno počáteční rozmístění dětí. Prostřední obrázek ukazuje výsledek následujícího přesazování: děti číslo 1 a 2 se posunou o jedno místo, děti číslo 3 a 5 se posunou o dvě místa, a děti číslo 4 a 6 zůstanou na svém původním místě. Toto rozesazení splňuje podmínky zadání, neboť dítě 3 sedí mezi dětmi 6 a 4, 4 sedí mezi 3 a 5, 5 sedí mezi 4 a 1, 1 sedí mezi 5 a 2, 2 sedí mezi 1 a 6, a 6 sedí mezi 2 a 3. Existuje ještě jedno možné konečné rozmístění dětí, znázorněné na obrázku vpravo. V obou případech se žádné z dětí neposune o víc než o dvě místa.

5. Hra s obdélníkem

Alice a Bajtazar hrají následující hru. Mají obdélník o rozměrech $x \times y$, kde x a y jsou kladná celá čísla. Hráči se pravidelně střídají na tahu. Tah spočívá v rozdělení obdélníka na dvě obdélníkové části jedním svislým nebo vodorovným řezem. Výsledné obdélníky musí mít kladné celočíselné rozměry.



Obr. 65. Možné řezy v obdélníku o rozměrech 4×3.

Menší ze vzniklých obdélníků je odstraněn a zbývající z nich je předán druhému hráči. Je-li obdélník rozdělen přesně na stejné poloviny, pak je odstraněna jedna z nich. Hráč, který dostane obdélník o rozměrech 1×1 prohrál, protože nemůže provést tah.

Alice v této hře neustále prohrává. Napište program, který jí proti Bajtazarovi pomůže vyhrát. Program musí používat speciální knihovnu, pomocí které bude hru hrát. Knihovna poskytuje funkce `dimension_x()` a `dimension_y()`, které vrací rozměry obdélníka. Počáteční rozměry obdélníka jsou celá čísla mezi 1 a 100 000 000. Alespoň jeden z rozměrů je větší než 1. Pro 50 % vstupních dat rozměry navíc nepřekročí 25.

Knihovna dále obsahuje proceduru `cut(dir, position)`, jejímž zavoláním váš program provede tah. Parametry `dir` a `position` popisují směr a umístění řezu. Parametr `dir` musí nabývat jedné z hodnot `vertical` nebo `horizontal`. Pokud `dir` = `vertical`, pak je řez veden svisle, parametr `position` určuje x-ovou souřadnici řezu (viz obrázek nahoře) a musí splňovat podmínku 1 ≦ `position` ≦ `dimension_x()` − 1. Pokud `dir` = `horizontal`, pak je řez vodorovný, parametr `position` určuje y-ovou souřadnici řezu a musí splňovat 1 ≦ `position` ≦ `dimension_y()` − 1.

Váš program bude hrát za prvního z hráčů — musí rozdělit počáteční obdélník. Když program zavolá proceduru `cut`, jeho tah je zaznamenán a řízení je předáno soupeři. Poté, co táhne soupeř, se řízení vrátí vašemu programu. Funkce `dimension_x()` a `dimension_y()` budou vracet rozměry obdélníka po provedení vašeho i soupeřova tahu. Váš program bude ukončen poté, co vyhraje, prohraje, nebo provede neplatný tah (tj. zavolá proceduru `cut` s chybnou hodnotou parametrů). Ukončení proběhne automaticky, váš program by tedy měl hrát tak dlouho, dokud je to možné. Můžete předpokládat, že pro každý z testovacích vstupů existuje vyhrávající strategie pro váš program.

Váš program nesmí číst z žádného souboru, ani zapisovat do žádného souboru, nesmí používat standardní vstup a výstup, a nesmí se pokusit měnit obsah paměti, která mu nepatří. Porušení kteréhokoliv z těchto pravidel může vést k diskvalifikaci.

6. Řeky

V Bajtánském království mají mnoho lesů a řek. Malé řeky se vlévají do větších, a nakonec se všechny řeky spojí v jeden mohutný tok, který u Bajtic ústí do moře.

V Bajtánii je n dřevorubeckých osad, ležících poblíž řek. V Bajticích se nachází Ústřední dřevařský závod, který zpracovává stromy, poražené v celém království. Stromy se dopravují do Bajtic po řekách. Doprava je však příliš drahá, a proto se král rozhodl nechat postavit k dalších dřevařských závodů v některých osadách. Stromy pak nebude nutné dopravovat až do Bajtic, ale bude možné zpracovat je v nejbližším závodě po proudu řeky. Stromy poražené poblíž osady s dřevařským závodem se po řece nebudou dopravovat vůbec. Je třeba podotknout, že bajtánské řeky se nerozvětvují, a z každé osady se tedy dá dostat právě jedním způsobem po proudu řeky do Bajtic.

Královští úředníci spočítali, kolik stromů se porazí u každé osady za rok. Máte rozhodnout, kde postavit dřevařské závody tak, aby celková cena za dopravu stromů po řece byla co nejmenší. Doprava jednoho stromu po řece stojí jeden cent za kilometr.

Úloha:

Napište program, který:

▷ načte ze standardního vstupu počet osad, počet stromů poražených za rok u každé z nich, počet nově budovaných dřevařských závodů a popis řek,

▷ určí nejmenší možnou cenu za dopravu stromů po řekách po postavení nových dřevařských závodů, a

▷ vypíše výsledek na standardní výstup.

Vstup: První řádek vstupu obsahuje dvě celá čísla oddělená mezerou: n — počet osad různých od Bajtic ($2 \leqq n \leqq 100$), a k — počet nových dřevařských závodů, které se mají vybudovat ($1 \leqq k \leqq 50$ a $k \leqq n$). Osady jsou očíslovány $1, 2, \ldots, n$, Bajtice jsou označeny číslem 0.

Každý z n následujících řádků obsahuje tři celá čísla, oddělená vždy jednou mezerou. Řádek $i + 1$ obsahuje:

▷ w_i — počet stromů poražených u osady číslo i za rok ($0 \leqq w_i \leqq 10\,000$),

▷ v_i — číslo první osady (nebo 0, pokud jsou to Bajtice) po proudu řeky od osady číslo i ($0 \leqq v_i \leqq n$),

▷ d_i — vzdálenost (v kilometrech) po proudu řeky od osady číslo i do osady v_i ($1 \leqq d_i \leqq 10\,000$).

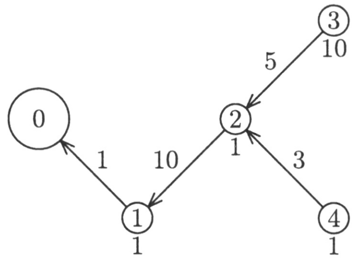
Celková cena za dopravu všech stromů do Bajtic za rok nepřesáhne 2 000 000 000 centů.

Pro 50 % vstupních dat navíc platí, že $n \leqq 20$.

Výstup: Program vypíše na standardní výstup právě jedno celé číslo — nejmenší možnou cenu za dopravu stromů po řekách (v centech).

Příklad vstupu a výstupu:

```
4 2 4
1 0 1
1 1 10
10 2 5
1 2 3
```



Obr. 66. Obrázek odpovídá vstupním datům z příkladu. Čísla osad jsou uvnitř kroužků. Čísla pod kroužky udávají počet stromů poražených poblíž osad. Čísla nad šipkami udávají délky řek. Dřevařské závody je třeba postavit v osadách číslo 2 a 3.