

57. ročník matematické olympiády na středních školách

20. mezinárodní olympiáda v informatice

In: Karel Horák (editor); Daniel Král' (editor); Martin Mareš (editor); Peter Novotný (editor); Jaromír Šimša (editor); Jaroslav Švrček (editor); Pavel Töpfer (editor): 57. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 2007/2008. 49. mezinárodní matematická olympiáda. 20. mezinárodní olympiáda v informatice. (Czech). Praha: Jednota českých matematiků a fyziků, 2010. pp. 183–198.

Persistent URL: <http://dml.cz/dmlcz/405163>

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>

20. mezinárodní olympiáda v informatice

Jubilejní 20. ročník Mezinárodní olympiády v informatice (IOI) se uskutečnil ve dnech 16.–23. srpna 2008 v egyptském hlavním městě Káhiře. Soutěže se zúčastnilo 283 soutěžících ze 78 zemí celého světa; nově vyslali své soutěžní týmy Alžírsko a Ghana. Většina zemí využila možnosti vyslat na IOI maximální povolený počet čtyř soutěžících, z několika nově se zúčastňujících zemí přijely reprezentace menší.



Reprezentační družstvo České republiky bylo sestaveno na základě výsledků dosažených soutěžícími v ústředním kole 57. ročníku Matematické olympiády – kategorie P (programování), které se uskutečnilo v dubnu 2008 v Českých Budějovicích. Naše družstvo pro IOI 2008 mělo následující složení:

Miroslav Klimoš, student gymnázia M. Koperníka v Bílovci,
Jan Matějka, student gymnázia Jírovcova v Českých Budějovicích,
Roman Smrž, absolvent gymnázia E. Krásnohorské v Praze,
Vojtěch Tůma, absolvent gymnázia J. Masaryka v Jihlavě.

Vedoucími české delegace byli jmenováni RNDr. *Daniel Král*, Ph.D., a Bc. *Tomáš Gavenčiak*, oba z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze. Vedle této oficiální šestičlenné delegace se 20. mezinárodní olympiády v informatice zúčastnil z České republiky ještě Mgr. *Martin Mareš*, Ph.D., rovněž pracovník MFF UK v Praze, a to jako člen Mezinárodního vědeckého výboru IOI.

Naši studenti měli možnost připravit se na soutěž nejen samostatným studiem, ale i účastí na přípravném soustředění, které tradičně pořádáme společně s polskými a slovenskými organizátory (CPSPC — Czech-Polish-Slovak Preparation Camp). Letos toto přípravné soustředění uspořádalo vedení slovenské olympiády v informatice ve slovenských Danišovcích. Soustředění se zúčastnili R. Smrž, V. Tůma a vybraní účastníci CEOI 2008; zbylí dva studenti vybraní pro účast na IOI 2008 se soustředění nemohli zúčastnit z časových důvodů.

Pořadatelé letošní Mezinárodní olympiády v informatice bylo egyptské ministerstvo školství a ministerstvo informačních a komunikačních technologií. Čestnou záštitu nad soutěží převzal egyptský prezident Husny Mubarak. Ubytování a stravování soutěžních týmů bylo zajištěno ve vzdělávacím komplexu Mubarak Education City v nově budované čtvrti káhirské metropole 6 October City. V tomto komplexu proběhla samotná soutěž i oficiální zahájení a ukončení soutěže, kterých se zúčastnili egyptský ministr školství a egyptský ministr informačních a komunikačních technologií.

Dnem příjezdu účastníků na soutěž byla stanovena sobota 16. srpna 2008. Český tým odlétal z Prahy kolem poledne a přiletěl do Káhiry v jedenáct hodin večer. Na místním letišti pořadatelé jednotlivé týmy očekávali a zajistili jim transport na místo soutěže. Soutěžními dny bylo pondělí 18. srpna 2008 a středa 20. srpna 2008. Zbylé dny pak byly vyhrazeny pro výběr úloh, zasedání mezinárodní jury a doprovodný společenský a kulturní program. V každém soutěžním dni byla účastníkům předložena sada tří úloh, na jejichž vyřešení měl každý soutěžící vymezen čas 5 hodin. Mezinárodní olympiáda v informatice je soutěží jednotlivců a tak každý soutěžící pracoval na přiděleném osobním počítači s předem oznámeným soutěžním prostředím. Vyhlášení soutěže proběhlo netradičně již ve čtvrtek 21. srpna 2008, neboť pátek a sobota jsou podle islámské tradice volné dny. Dnem odjezdu dle programu soutěže pak byla sobota 23. srpna 2008; česká delegace využila jeden z prvních ranních letů z Káhiry, který umožnil mimopražským účastníkům soutěže bezproblémový odjezd domů.

Během Mezinárodní olympiády v informatice proběhla na zasedání jury drobná revize pravidel soutěže. Jako nové pravidlo bylo schváleno, že student může reprezentovat na soutěži buď zemi, kde studuje střední školu, nebo (nově) zemi, jejímž je občanem. Vedoucí týmů byli informováni o výdajích na propagaci soutěže. Registrační poplatek na soutěž byl ponechán ve stejné výši, která je 200 EUR nebo 300 USD. Podstatná část vybraných poplatků je čerpána na rozvoj webových stránek soutěže, které jsou spravovány na kanadské University of Waterloo, workshopy určené k testování nových typů soutěžních úloh a podporu konference o mezinárodních soutěžích v informatice, jejíž druhý ročník se uskutečnil letos společně s Mezinárodní olympiádou v informatice.

Mezinárodní výbor soutěže, který nad celou soutěží bdí, pak vedoucí týmů informoval o problémech s reprezentací Chile na soutěži, neboť chilské ministerstvo školství oficiálně na soutěž vyslalo jiný tým, než který se

soutěže zúčastnil. Celou situaci řeší mezinárodní výbor soutěže ve spolupráci s prezidentem soutěže. Vedoucí delegací byly též informováni, že Mezinárodní olympiáda v informatice se v roce 2012 uskuteční v Itálii, která jako jediná země projevila zájem olympiádu v tomto roce uspořádat.

Vraťme se nyní k průběhu soutěže. Jak již bylo zmíněno, soutěžící řešili v každém soutěžním dni sadu tří úloh, na jejichž vyřešení byl stanoven časový limit 5 hodin. Letošní sada úloh, na jejichž přípravě se letos výraznou měrou podíleli členové Mezinárodního vědeckého výboru soutěže, byla velmi obtížná, o čemž svědčí fakt, že pouze 15 soutěžících (z 283) získalo více než 400 bodů z možných 600 bodů. Úlohy však byly připraveny velmi kvalitně a jejich popis byl velmi výstižný. Během soutěže se k zadání úloh vyskytlo méně než 10 dotazů (počet dotazů bývá obvykle v řádu desítek). Řešení odevzdaná soutěžícími byla testována pomocí softwaru vyvinutého pro polskou olympiádu v informatice na předem připravených testovacích datech se stanovenými časovými limity. Použití časových limitů umožňuje odlišit kvalitu (rychlost) použitého algoritmu. Zadání úloh zaručuje, že i neoptimální řešení získají jisté množství bodů. Proti vyhodnocení úloh se neobjevily žádné protesty.

Podle pravidel Mezinárodní olympiády v informatice obdrží polovina soutěžících medaili a počty zlatých, stříbrných a bronzových medailí se určí tak, aby jejich počet byl v poměru 1 : 2 : 3. Letos tedy bylo uděleno 24 zlatých, 47 stříbrných a 70 bronzových medailí. O již zmiňované obtížnosti úloh svědčí fakt, že k získání bronzové medaile stačilo pouhých 127 bodů ze 600 možných a 26 soutěžících nezískalo vůbec žádný bod. I přes zmiňovanou obtížnost úloh se dvěma našim soutěžícím podařilo získat stříbrné medaile. Výsledky našich studentů byly následující:

47.	Miroslav Klimoš	267 bodů	stříbrná medaile
71.	Roman Smrž	229 bodů	stříbrná medaile
197.	Jan Matějka	53 bodů	
219.	Vojtěch Tůma	33 bodů	

Mezinárodní olympiáda v informatice je soutěží jednotlivců; žádné oficiální pořadí zemí se neurčuje. Z hlediska počtu a kvality získaných medailí bychom se v hodnocení národních delegací umístili přibližně v druhé čtvrtině zúčastněných zemí; nejúspěšnějšími zeměmi byly již tradičně ČLR a Polsko (se třemi zlatými a jednou stříbrnou medailí) a Rusko

a USA (se dvěma zlatými a dvěma stříbrnými medailemi). Soutěžícím ze Slovenska se tentokrát vedlo o něco lépe než nám, získali v soutěži dvě stříbrné a dvě bronzové medaile. Výsledky našich soutěžících napovídají, že předchozí zkušenosti účastníků soutěže mají výrazný vliv na jejich umístění (M. Klimoš a R. Smrž se soutěže IOI zúčastnili již vloni). Podrobnější informace o soutěži, texty soutěžních úloh a celkové výsledky všech účastníků lze nalézt na Internetu na adrese

<http://www.ioi2008.org/>.

Následující, v pořadí 21. ročník IOI se bude konat v srpnu 2008 v bulharském Plovdivu. Tamní organizátoři již vyzvali všechny země zúčastněné na IOI v Egyptě k účasti v příštím ročníku soutěže. Byla též stanovena i místa konání několika dalších ročníků IOI: 2010 Kanada, 2011 Thajsko a 2012 Itálie.

Texty soutěžních úloh

1. Jednorozměrná zahrada

Faraón Ramses II. se právě vrátil z další prohrané bitvy. Na věčnou připomínku své čestné porážky se rozhodl postavit smuteční zahradu, která spojí jeho palác v Luxoru s chrámem v Karnaku. V zahradě budou pěstovány pouze květy lotosu a papyrové rákosí, tedy rostliny, které symbolizují Horní a Dolní Egypt.

Zahrada bude rozdělena na N částí a v každé z nich se bude pěstovat jedna ze dvou vybraných rostlin. Jako symbol rovného postavení Horního a Dolního Egypta musí být zahrada *vyvážená* v následujícím smyslu: v každé posloupnosti po sobě následujících částí zahrady se počet částí s lotosovými květy liší od počtu částí s papyrovým rákosím nejvýše o 2.

Osazení jednotlivých částí zahrady rostlinami lze reprezentovat jako řetězec N písmen ‚L‘ a ‚P‘, která představují části osazené lotosovými květy a papyrovým rákosím. Pro $N = 5$ existuje celkem 14 *vyvážených* osazení zahrady: LLPLP, LLPPL, LPLLP, LPLPL, LPLPP, LPPLL, LPPLP, PLLPL, PLLPP, PLPLL, PLPLP, PLPPL, PPLLP a PPLPL.

Řetězce kódující *vyvážená* osazení zahrady rostlinami lze seřadit abecedně a poté očíslovat celými čísly od 1. Např. pro $N = 5$ má řetězec PLPPL pořadové číslo 12.

Úkol: Vaším úkolem je napsat program, který pro zadaný počet N částí zahrady a zadaný řetězec představující osazení zahrady rostlinami

vypočte *zbytek* pořadového čísla tohoto řetězce *po dělení* zadaným číslem M . Hodnota čísla M nemá žádný vliv na řešení úlohy kromě zjednodušení výpočtů.

Omezení:

$1 \leq N \leq 1\,000\,000$ počet částí zahrady

$7 \leq M \leq 10\,000\,000$ číslo M

Bodování: U testovacích vstupů v hodnotě alespoň 40 bodů je počet N částí zahrady nejvýše 40.

Popis vstupu: Váš program načte ze standardního vstupu data v následujícím tvaru:

- ▷ První řádek obsahuje celé číslo N , které udává počet částí zahrady.
- ▷ Druhý řádek obsahuje celé číslo M .
- ▷ Třetí řádek obsahuje řetězec N znaků ‚L‘ (lotos) a ‚P‘ (papyrus), který představuje *vyvážené* osazení zahrady.

Popis výstupu: Váš program zapíše na standardní výstup jedno celé číslo mezi 0 a $M - 1$ (včetně), které je *zbytek* pořadového čísla zadaného řetězce *po vydělení* číslem M .

Příklad 1:

Vzorový vstup 1 Vzorový výstup 1

5 5

7

PLPPL

Číslo přiřazené řetězci PLPPL je 12 a jeho zbytek po dělení 7 je 5.

Příklad 2:

Vzorový vstup 2 Vzorový výstup 2

12 39

10000

LPLLPLPPLLL

2. Ryby

Čtyřicáté druhé noci princezna Šeherezáda vyprávěla pohádku o dalekém jezeře uprostřed pouště, ve kterém žilo F vzácných ryb. Z nejvzácnějších drahokamů světa bylo vybráno K různých druhů a každá z ryb spolkla jeden drahokam. Vyprávěla, že druhů drahokamů (K) možná bylo méně než ryb v jezeře (F), a tak více různých ryb mohlo spolknout tentýž druh drahokamu.

Jak čas běžel, větší ryby pozřely některé z menších ryb v jezeře. Ryba může sníst jinou rybu právě tehdy, když je alespoň dvakrát delší (ryba A

může sníst rybu B , právě když $L_A \geq 2L_B$). Kromě tohoto pravidla není vzájemné požírání ryb nijak omezeno. Jedna ryba může postupně sníst několik menších ryb nebo také nemusí sníst žádnou rybu, ačkoliv by mohla. Pokud větší ryba sní menší, její délka se nezmění a drahokamy, které byly původně v žaludku menší ryby, přibudou k těm v žaludku větší ryby.

Šeherezáda vyprávěla, že kdokoliv najde toto jezero, může z něj jednu rybu vylovit a obsah jejího žaludku si ponechat. Rádi byste také zkusili své štěstí, ale ještě než se vydáte na cestu, by vás zajímalo, kolik různých kombinací drahokamů může být v žaludku vylovené ryby.

Úkol: Vaším úkolem je napsat program, který ze znalosti délek ryb, které v jezeře původně žily, a obsahu jejich žaludků určí počet různých kombinací drahokamů, které se mohou vyskytnout v žaludku nějaké ryby, modulo zadané celé číslo M . Kombinace drahokamů je jednoznačně popsána počtem drahokamů každého z K druhů. Druhy drahokamů jsou vzájemně neporovnatelné jak krásou tak hodnotou a drahokamy téhož druhu jsou nerozlišitelné.

Omezení:

$1 \leq F \leq 500\,000$	počet ryb, které původně žily v jezeře
$1 \leq K \leq F$	počet druhů drahokamů
$2 \leq M \leq 30\,000$	číslo M

Popis vstupu: Váš program načte ze standardního vstupu data v následujícím tvaru:

- ▷ První řádek obsahuje celé číslo F , které udává původní počet ryb.
- ▷ Druhý řádek obsahuje celé číslo K , které určuje počet druhů drahokamů. Druhy drahokamů jsou očíslovány čísly od 1 do K (včetně).
- ▷ Třetí řádek obsahuje číslo M .
- ▷ Každý z následujících F řádků popisuje jednu z původních ryb. Řádek obsahuje dvě celá čísla oddělená jednou mezerou. První udává délku ryby a druhé určuje druh drahokamu, který ryba na začátku pohádky spolkla.

Poznámka: Je zaručeno, že pro každý z K druhů drahokamů existuje ryba, která ho má ve svém žaludku.

Popis výstupu: Váš program zapíše na standardní výstup jedno číslo mezi 0 a $M - 1$ (včetně), které je zbytkem po dělení počtu možných kombinací drahokamů v žaludku vylovené ryby číslem M . Hodnota čísla M nemá na řešení úlohy žádný vliv kromě zjednodušení výpočtů.

Bodování: U testovacích vstupů v hodnotě alespoň 70 bodů hodnota K nepřekročí 7 000. A také u testovacích vstupů v hodnotě alespoň 25 bodů hodnota K nepřekročí 20.

Příklad:

<i>Vzorový vstup</i>	<i>Vzorový výstup</i>
5	4
3	
7	
2 2	
5 1	
8 3	
4 1	
2 3	

Existuje celkem 11 kombinací drahokamů, které by se mohly vyskytnout v žaludku vylovené ryby. Váš program by tedy měl vypsát číslo 4, které je zbytkem 11 po dělení 7. Možné kombinace jsou následující: [1] [1, 2] [1, 2, 3] [1, 2, 3, 3] [1, 3] [1, 3, 3] [2] [2, 3] [2, 3, 3] [3] a [3, 3], kde např. [2, 3, 3] znamená, že žaludek vylovené ryby obsahuje jeden drahokam druhu č. 2 a dva drahokamy druhu č. 3.

Popišme si, jak je lze tyto kombinace dosáhnout:

- ▷ [1]: Je možné, že vylovíte druhou nebo čtvrtou rybu před tím, než sníte nějakou jinou.
- ▷ [1, 2]: Je možné, že vylovíte druhou rybu poté, co snědla první.
- ▷ [1, 2, 3]: Jednou z možností je vylovit třetí rybu, potom co snědla čtvrtou rybu, která už před tím snědla první.
- ▷ [1, 2, 3, 3]: Čtvrtá sní první, třetí sní čtvrtou, třetí sní pátou a vylovíte třetí (a sníte ji).
- ▷ [1, 3]: Chytíte třetí, která snědla čtvrtou.
- ▷ [1, 3, 3]: Třetí sní čtvrtou a pátou.
- ▷ [2]: První ryba.
- ▷ [2, 3]: Třetí sní první.
- ▷ [2, 3, 3]: Třetí sní první a pátou.
- ▷ [3]: Třetí ryba.
- ▷ [3, 3]: Třetí sní pátou.

3. Ostrovy

Rozhodli jste se navštívit zábavný zábavní park rozkládající se na N ostrovech. Při budování parku začali z každého ostrova stavět je-

den most na některý jiný ostrov. Po dokončení stavby je tedy v parku N mostů a na některé ostrovy vede více mostů. Délka mostu stavěného z i -tého ostrova je L_i . Každý z mostů lze používat v obou směrech. Kromě toho v parku také funguje mezi každou dvojicí ostrovů přívoz.

Protože jste velmi rychle zjistili, že nejzábavnější atrakcí v parku je chůze po dlouhatánských mostech, rádi byste po nich v parku ušli co největší vzdálenost. Při vašem pohybu v parku se však chcete řídit následujícími pravidly:

- ▷ Svou návštěvu můžete začít na vámi vybraném ostrově.
- ▷ Žádný ostrov nenavštívíte více než jednou.
- ▷ Z ostrova A se můžete (přímo) přesunout na ostrov B , který jste dosud nenavštívili, a to jedním ze dvou následujících způsobů:
 - ▷ Chůzí po mostě, pokud mezi ostrovy A a B vede most. V tomto případě je délka tohoto mostu (v případě dvou souběžných mostů délka delšího z nich) je přičtena k celkové vzdálenosti, kterou jste už ušli po mostech.
 - ▷ Přívozem, pokud ostrov B není dosažitelný z ostrova A pomocí mostů a přívozů, které jste již během své návštěvy parku použili. (Při rozhodování o dosažitelnosti ostrova B z A uvažujte i cesty vedoucí přes již navštívené ostrovy.)

Vášim úkolem není navštívit všechny ostrovy. Podobně nemusí být možné projít přes všechny mosty.

Úkol: Vaším úkolem je napsat program, který z popisu N mostů v parku určí nejdelší možnou vzdálenost, kterou lze po mostech ujít při dodržení výše uvedených podmínek.

Omezení:

- $2 \leq N \leq 1\,000\,000$ počet ostrovů v parku
 $1 \leq L_i \leq 100\,000\,000$ délka mostu vystavěného z i -tého ostrova

Popis vstupu: Váš program načte ze standardního vstupu data v následujícím tvaru:

- ▷ První řádek obsahuje celé číslo N , které udává počet ostrovů v parku. Ostrovy jsou očíslovány čísly od 1 do N (včetně).
- ▷ Každý z následujících N řádků popisuje jeden z mostů; i -tý z těchto řádků obsahuje dvě celá čísla oddělená jednou mezerou a popisuje most vystavěný z ostrova číslo i . První z těchto dvou čísel je číslo ostrova, kam most vede, druhé je jeho délka L_i . Můžete předpokládat, že každý most spojuje dva různé ostrovy.

Popis výstupu: Váš program zapíše na standardní výstup (na jedi-

ném řádku) jedno číslo, které udává největší možnou vzdálenost, kterou můžete v parku po mostech ujít.

Poznámky: 1. Pro některé testovací vstupy výsledná hodnota překročí rozsah 32bitového celočíselného typu a k získání plného počtu bodů za tuto úlohu může být nutné použít 64bitové typy (int64 v Pascalu a long long v C/C++).

2. Programy v jazyce Pascal, které používají k načtení vstupu 64bitové datové typy, jsou výrazně pomalejší než stejné programy používající 32bitové typy a to i tehdy, když načítané hodnoty nepřekročí rozsah 32bitových celočíselných typů. Doporučujeme tedy k načtení vstupu používat pouze 32bitové celočíselné typy.

Bodování: U testovacích vstupů v hodnotě alespoň 40 bodů hodnota N nepřekročí 4 000.

Příklad:

Vzorový vstup

Vzorový výstup

7

24

3 8

7 2

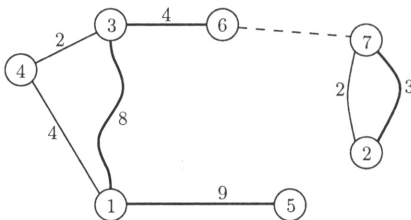
4 2

1 4

1 9

3 4

2 3



$N = 7$ mostů v příkladu spojuje dvojice ostrovů (1–3), (2–7), (3–4), (4–1), (5–1), (6–3) a (7–2). Pověšimněte si, že ostrovy 2 a 7 jsou spojeny dvěma různými (souběžnými) mosty.

Jedna z možností, jak ujít maximální možnou vzdálenost, je:

- ▷ Začněte na ostrově 5.
- ▷ Přejděte po mostě délky 9 na ostrov 1.
- ▷ Přejděte po mostě délky 8 na ostrov 3.
- ▷ Přejděte po mostě délky 4 na ostrov 6.
- ▷ Použijte přívoz z ostrova 6 na ostrov 7.
- ▷ Přejděte po mostě délky 3 na ostrov 2.

Svou cestu zakončíte na ostrově 2 a celkem jste po mostech ušli $9 + 8 + 4 + 3 = 24$. Jediný nenavštívený ostrov má číslo 4. Všimněte si, že na konci cesty už tento ostrov navštívit nemůžete:

- ▷ Nelze se na něj z ostrova 2 dostat přímo po mostě.

- ▷ Nesmíte na něj přejet přívozem, neboť ostrov 4 je *dosažitelný* z ostrova 2 následujícím způsobem: Přešli byste zpět po mostě na ostrov 7, pak byste pokračovali již jednou použitým přívozem na ostrov 6 a odtud byste došli po mostech na ostrov 4.

4. Knihťisk

Je potřeba vytisknout N slov pomocí starého jednořádkového knihařského lisu. Tento typ lisu používá k tisku kovové litery, tj. malé kousky kovu každý s jedním písmenem. Litery se skládají za sebou a vytvoří tak jedno slovo, které lze pak pomocí lisu obtisknout na papír. Litery lze do lisu přidávat a odebírat pouze na konci vytvářeného slova. S vytvářeným slovem je tedy možné provádět pouze následující *operace*:

- ▷ přidání jednoho písmene (litery) na konec slova v lisu,
- ▷ odebrání jednoho písmene (litery) z konce slova v lisu,
- ▷ vytištění slova v lisu.

Na začátku lis neobsahuje žádné litery. Na konci tisku je možné v lisu zanechat libovolné množství liter. Zadaná slova můžete vytisknout v libovolném pořadí.

Snažíte se vytisknout všechna zadaná slova co nejrychleji, tedy pomocí nejmenšího možného počtu výše uvedených *operací*.

Úkol: Vaším úkolem je napsat program, který pro zadaných N slov, určí nejmenší možný počet *operací* nutných k vytištění zadaných slov a jednu takovou posloupnost *operací* vypíše.

Omezení: $1 \leq N \leq 25\,000$ maximální možný počet zadaných slov

Popis vstupu: Váš program načte ze standardního vstupu data v následujícím tvaru:

- ▷ První řádek obsahuje celé číslo N , které udává počet zadaných slov
- ▷ Každý z následujících N řádků obsahuje jedno zadané slovo. Slovo se skládá pouze z malých písmen anglické abecedy ($'a'-'z'$) a jeho délka je alespoň 1 a nejvýše 20.

Všechna zadaná slova jsou navzájem různá.

Popis výstupu: Váš program запиše své výsledky na standardní výstup v následujícím tvaru:

- ▷ První řádek bude obsahovat jedno celé číslo M , které udává nejmenší možný počet *operací* potřebných k vytištění zadaných slov.
- ▷ Každý z následujících M řádků bude obsahovat právě jeden znak, který odpovídá jedné z *operací* následujícím způsobem:

- ⇒ Přidání písmene na konec slova v lisu je reprezentováno přidávaným písmenem (malé anglické abecedy),
- ⇒ Odebrání posledního písmene slova v lisu je reprezentováno znakem `,` (mínus, ASCII kód 45)
- ⇒ *Operace* vytištění slova v lisu je reprezentována znakem `,P` (velké písmeno *pé*).

Posloupnost *operací* na výstupu musí mít délku M a vést k vytištění všech zadaných slov.

Bodování: U testovacích vstupů v hodnotě alespoň 40 bodů hodnota N nepřekročí 18.

Příklad:

<i>Vzorový vstup</i>	<i>Vzorový výstup</i>
----------------------	-----------------------

3	20
---	----

print	t
-------	---

the	h
-----	---

poem	e
------	---

P

-

-

-

P

o

e

m

P

-

-

-

r

i

n

t

P

5. Teleporty

Účastníte se velkého závodu napříč Egyptem podél rovnoběžky směrem ze západu na východ. Závod má jen dvě pravidla: musíte se pohybovat pouze po rovnoběžce, a to vždy pouze na východ.

Na trati je umístěno N teleportů. Každý teleport má dva konce. Jakmile dorazíte k jednomu z jeho konců, teleport vás okamžitě nasosne a přenesení na svůj druhý konec. Teleport vás přemístí směrem na východ nebo na západ podle toho, ke kterému jeho konci dorazíte. Přesuny teleportem směrem na západ neodporují pravidlům závodu. Po teleportaci však musíte pokračovat dál na východ, jak určují pravidla závodu. Teleportaci se nemůžete nijak vyhnout. Na každém místě trasy závodu se vyskytuje konec nejvýše jednoho z teleportů; každý teleport spojuje dvě různá místa na trase. Konce teleportů jsou pouze mezi startem a cílem závodu (mimo start a cíl).

Pokaždé, když se teleportujete, získáte jeden bod. Cílem závodu je získat největší možný počet bodů. Abyste získali co nejvíce bodů, smíte před začátkem závodu postavit na trasu závodu M nových teleportů. Konce těchto teleportů můžete umístit kamkoliv mezi start a cíl závodu (mimo start a cíl) tak, aby se stále na každém místě nacházel nejvýše jeden konec teleportu a každý teleport spojoval dvě různá místa. Konce nových teleportů mohou být i v *neceločíselných* vzdálenostech od startu. Body získáte i za použití nově postavených teleportů.

Je zaručeno, že se vždy dostanete do cíle závodu, ať se děje, co se děje, tj. po jakémkoliv přidání M nových teleportů se chůzí na východ dostanete ze startu závodu do jeho cíle.

Úkol: Vaším úkolem je napsat program, který pro N zadaných teleportů a zadaný největší počet M teleportů, které můžete postavit, vypočte největší možný počet bodů, které můžete během závodu získat.

Omezení:

- | | |
|--|--|
| $1 \leq N \leq 1\,000\,000$ | původní počet teleportů na trase závodu |
| $1 \leq M \leq 1\,000\,000$ | největší možný počet teleportů,
které můžete postavit |
| $1 \leq W_i \leq E_i \leq 2\,000\,000$ | vzdálenosti konců i -tého teleportu od startu |

Popis vstupu: Váš program načte ze standardního vstupu data v následujícím tvaru:

- ▷ První řádek obsahuje celé číslo N , které udává původní počet teleportů.
- ▷ Druhý řádek obsahuje celé číslo M , které určuje počet teleportů, které můžete postavit.
- ▷ Každý z následujících N řádků popisuje jeden z původních teleportů. Řádek obsahuje dvě celá čísla W_i a E_i oddělená jednou mezerou. Tato čísla udávají polohu západního a východního konce teleportu

jako vzdálenost od startu závodu. První z těchto dvou čísel je vždy menší než druhé.

Je zaručeno, že žádné dva konce teleportů nemají stejnou vzdálenost od startu. Start závodu se nachází na pozici 0 a cíl je ve vzdálenosti 2 000 001 od startu závodu.

Popis výstupu: Váš program zapíše na standardní výstup jedno celé číslo, které udává největší možný počet bodů, které můžete v závodě získat.

Bodování: Pro testovací vstupy v hodnotě alespoň 30 bodů platí, že $N \leq 500$ a $M \leq 500$.

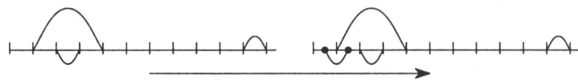
Příklad 1:

Vzorový vstup 1

3
1
10 11
1 4
2 3

Vzorový výstup 1

6



Na obrázku vlevo je počáteční úsek trati závodu s původními teleporty a vpravo je tentýž úsek s nově postaveným teleportem, který spojuje místa ve vzdálenosti 0,5 a 1,5 od startu závodu.

Po přidání nového teleportu způsobem znázorněným na obrázku bude vaše cesta ze startu do cíle vypadat takto:

- ▷ Začnete na pozici 0 (start) a vydáte se na východ.
- ▷ Ve vzdálenosti 0,5 od startu vás nově postavený teleport přenese do vzdálenosti 1,5 a tím získáte jeden bod.
- ▷ Pokračujete na východ a dosáhnete konce teleportu ve vzdálenosti 2 od startu. Teleport vás přenese do vzdálenosti 3 od startu a získáte druhý bod.
- ▷ Po dosažení konce teleportu ve vzdálenosti 4 od startu, jste teleportováni do vzdálenosti 1. Získali jste třetí bod.
- ▷ Ve vzdálenosti 1,5 od startu vás nově postavený teleport přenese do vzdálenosti 0,5 a máte již 4 body.
- ▷ Ve vzdálenosti 1 od startu jste teleportováni do vzdálenosti 4. Získali jste už 5 bodů.
- ▷ Ve vzdálenosti 10 od startu jste teleportováni do vzdálenosti 11. Nyní máte 6 bodů.
- ▷ Pokračujete na východ až do cíle závodu. Celkem jste získali 6 bodů.

Příklad 2:

Vzorový vstup 2

3

3

5 7

6 10

1999999 2000000

Vzorový výstup 2

12

6. Stavba pyramidy

Veliký faraón Ramses XLII. vás pověřil nalezením co největšího stavebního pozemku pro jeho pyramidu. K dispozici vám jeho úředníci dali územní plány, které popisují krajinu jako mřížku M krát N čtvercových polí. Základna pyramidy musí být čtvercová oblast bez překážek se stranami rovnoběžnými s hranami mřížky.

V plánu je vyznačeno P překážek spolu s cenami za odstranění každé z nich. Odstranění i -té překážky stojí C_i . Každá z těchto překážek je obdélník se stranami rovnoběžnými s mřížkou. Překážky se mohou navzájem překrývat. Každá překážka zabírá celou plochu všech polí, na kterých leží. Základnu pyramidy je možné umístit pouze na políčka bez překážek. Kdykoli se rozhodnete odstranit překážku, musíte ji odstranit celou, tj. není možné odstranit pouze část překážky. Odstranění jedné překážky nijak neovlivní ostatní překážky (ani ty, které se s odstraňovanou překážkou překrývají).

Stavebního materiálu i pracovní síly na stavbu pyramidy má faraón dostatek, avšak odstranění překážek a úprava terénu jsou často velmi nákladné. Faraón proto určil, že celkový rozpočet na odstranění překážek nesmí přesáhnout B .

Úkol: Vaším úkolem je napsat program, který ze znalosti umístění P překážek a parametrů M , N a B určí největší možnou velikost strany základny pyramidy, na kterou si můžete dovolit vyčistit stavební pozemek.

Omezení a bodování: Váš program bude spuštěn na třech sadách testovacích vstupů. Ve všech testech bude splněna následující omezení:

$$\begin{array}{ll} 1 \leq M, N \leq 1\,000\,000 & \text{rozměry mřížky} \\ 1 \leq C_i \leq 7\,000 & \text{cena za odstranění } i\text{-té překážky} \\ 1 \leq X_{i1} \leq X_{i2} \leq M & x\text{-ové souřadnice krajních polí } i\text{-té překážky} \\ 1 \leq Y_{i1} \leq Y_{i2} \leq N & y\text{-ové souřadnice krajních polí } i\text{-té překážky} \end{array}$$

V první sadě testovacích vstupů v hodnotě 35 bodů platí:

$B = 0$ váš rozpočet (nesmíte odstranit žádné překážky)
 $1 \leq P \leq 1\,000$ počet překážek

Ve druhé sadě testovacích vstupů v hodnotě 35 bodů platí:

$0 \leq B \leq 2\,000\,000\,000$ váš rozpočet
 $1 \leq P \leq 30\,000$ počet překážek

Ve třetí sadě testovacích vstupů v hodnotě 30 bodů platí:

$B = 0$ váš rozpočet (nesmíte odstranit žádné překážky)
 $1 \leq P \leq 400\,000$ počet překážek

Popis vstupu: Váš program ze standardního vstupu načte data v následujícím tvaru:

- ▷ První řádek obsahuje dvě celá čísla M a N oddělená jednou mezerou.
- ▷ Druhý řádek obsahuje celé číslo B , které udává váš rozpočet.
- ▷ Třetí řádek obsahuje počet překážek P .
- ▷ Každý z následujících P řádků popisuje jednu překážku. Na i -tém z těchto řádků se nachází pět celých čísel oddělených od sebe jednou mezerou: $X_{i1}, Y_{i1}, X_{i2}, Y_{i2}$ a C_i (v tomto pořadí). Udávají postupně souřadnice nejnižšího nejlevějšího pole překážky, nejhornějšího nejpravějšího pole překážky a cenu odstranění překážky.

Pole v levém dolním rohu mřížky má souřadnice $(1, 1)$ a pole v pravém horním rohu má souřadnice (M, N) .

Popis výstupu: Váš program na standardní výstup vypíše jediný řádek obsahující jedno celé číslo. Toto číslo je největší možná délka strany základny pyramidy, pro kterou je možné připravit pozemek bez překážek při dodržení stanoveného rozpočtu. Pokud není možné postavit žádnou pyramidu, váš program vypíše 0.

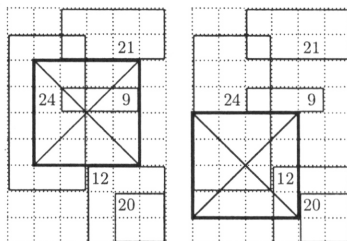
Příklad 1:

Vzorový vstup 1

```
6 9
42
5
4 1 6 3 12
3 6 5 6 9
1 3 3 8 24
3 8 6 9 21
5 1 6 2 20
```

Vzorový výstup 1

4



Na obrázku vidíte dvě možná umístění základny pyramidy, v obou případech o délce strany 4.

Příklad 2:

Vzorový vstup 2

13 5

0

8

8 4 10 4 1

4 3 4 4 1

10 2 12 2 2

8 2 8 4 3

2 4 6 4 5

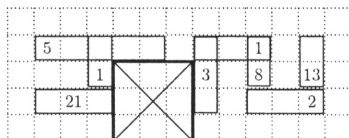
10 3 10 4 8

12 3 12 4 13

2 2 4 2 21

Vzorový výstup 2

3



Na obrázku vidíte jediné možné umístění základny pyramidy o délce strany 3.

