

fiks!



ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

FIT

Fitácký Informatický Korespondenční Seminář

Ročník 2018/2019, 2. kolo

Co je to FIKS?

FIKS je Fitácký Informatický Korespondenční Seminář pro středoškolské studenty pořádaný Fakultou informačních technologií ČVUT v Praze. Byl založen na podzim roku 2013 a nyní tak probíhá čtvrtý ročník (samozřejmě číslujeme od nuly). Nabízí možnost potrápít tvůj mozek řešením algoritmických úloh různé obtížnosti, od snadných po zapeklité, na nichž se můžeš leccos nového naučit a podstatně se zdokonalit.

Jak to probíhá?

Jeden ročník se skládá z několika kol a následného soustředění pro nejlepší řešitele. V těchto kolech, která trvají vždy přibližně dva měsíce, máš možnost v teple domova řešit zadané úlohy, a své řešení nám potom odešleš. My ti toto řešení opravíme, obodujeme a pošleme zpět, aby ses mohl poučit ze svých chyb. Spolu s tím zveřejníme vzorové řešení, které můžeš prostudovat a třeba se něco přiučit. Získané body se sčítají do konečného žebříčku, ze kterého vybereme ty nejlepší a pozveme je na již zmíněné soustředění.

Proč řešit FIKS?

Řešením každého problému, se kterým se potýkáme, se zdokonalujeme. Zde ti nabízíme možnost pořádně se zamyslet nad zajímavými algoritmickými problémy, vyzkoušet své algoritmické myšlení a programátorské dovednosti a naučit se spoustu nových věcí.

Také je to možnost seznámení s novými lidmi, které baví informatika, programování, matematika a přemýšlení vůbec. Těm nejlepším jsme schopni garantovat přijetí na FIT ČVUT bez přijímacích zkoušek.

Jak se můžu zapojit?

Začni nejprve tím, že se zaregistruješ na našich webových stránkách na adrese <http://fiks.fit.cvut.cz>. Potom si stáhni zadání úloh (nebo využij tuto brožurku), vyřeš je a své řešení nám tamtéž odevzdej.

Typy úloh

Celkem se ve FIKSu můžeš setkat se třemi typy úloh. O který typ úlohy se jedná, je vždy uvedeno u konkrétního zadání úlohy.

Nejčastěji se u nás potkáš s úlohami typu *Rozmysli, popiš a naprogramuj*. U každé úlohy tohoto typu se odevzdává jak popis algoritmu (s odhadem asymptotické složitosti), tak i zdrojový kód řešení problému v tebou zvoleném jazyce (jakýkoliv vyšší programovací jazyk dle tvé volby, například C, Java, Pascal, apod.).

Dalším typem jsou úlohy *Zamysli se*. Tyto úlohy jsou obvykle více teoretické a vyžadují, aby ses nad nimi důkladně zamyslel. Oproti předchozímu typu úloh nemusíš nic programovat, odevzdává se pouze slovní popis řešení problému.

Pokud nemáš rád teoretické úlohy a raději by sis procvičil/a své programátorské umění, pak pro je pro tebe určena kategorie *Odpověz Sfinze*. V úlohách tohoto typu po tobě nechceme popis algoritmu, je však potřeba vyřešit daný problém a toto řešení pak precizně naprogramovat. Oproti ostatním typům úloh se navíc okamžitě dozvíš, zda je tvé řešení správné, protože ho můžeš okamžitě odevzdat do našeho vyhodnocovacího systému.

Další a podrobnější informace nalezneš na našich webových stránkách.

Milý řešiteli FIKSu!

Letošní ročník FIKSu je v plném proudu a ty právě čteš brožurku se zadáními druhého kola. Pokud jsi nový řešitel a ještě jsi se nezapojil, tak nezoufej, do 31.1. je ještě možnost řešit úlohy prvního kola, ale i bez něj je stále možnost za řešení úloh získat spoustu bodů.

Před řešením úloh si nezapomeň přečíst příběh, do kterého přibyla další kapitola o Eleanořiných dobrodružstvích. Tentokrát bude muset Eleanor prokázat, že je schopná přidat se k odboji proti režimu.

Přejeme ti hodně zábavy při řešení úloh. V případě jakýchkoliv otázek nám neváhej napsat!

Tvoji organizátoři

Fitácký Informatický Korespondenční Seminář

Ročník 2018/2019, 2. kolo

Začátek kola: 24. 1. 2019 16:00
Termín odevzdání: 28. 2. 2019 23:59
Odevzdávání: Přes webové rozhraní na <http://fiks.fit.cvut.cz>
Další informace: <http://fiks.fit.cvut.cz>
kontakt@fiks.fit.cvut.cz

fiks!

Kapitola 0

Píše se rok 2056. Po skončení Velké války algoritmů (známá také jako Třetí světová kyber-válka) v roce 2039, je svět sjednocen pod jeden striktní režim, který potlačuje veškeré náznaky efektivních algoritmů, optimalizací a jiných fint (bitset). Je obecně známé, že tyto metody mohly za nadměrně rychlý vývoj technologií, zapříčinily závod ve zbrojení, a tak i vznik války. Používání efektivních algoritmů, či jejich pouhé psaní, je počín trestaný smrtí. Režim funguje zejména díky svým úřadům pro potírání algoritmů. Nejdůležitější z nich je FIT, který se zabývá vyhledáváním zmínek o efektivních algoritmech, jejich následnou cenzuru a předávání viníků orgánům činným v potíracím řízení.

Hrstka lidí, kteří nahlíží na systém nelibně, se seskupila do odbojové organizace FIKS. Snaží se o zachování efektivních algoritmů a jejich velkým cílem je osvobodit lid od systému. Fiksáci (název pro členy odboje) jsou systémem vyhledávání a bez milosti likvidováni, tak musí operovat ve stínech, a není známé, kdo ho vede, ani jakou mají organizaci.

Eleanor Sechsterer se narodila během války v roce 2037 v Praze. V roce 2040 byli oba její rodiče zajati a následně odsouzeni režimem pro jejich působení ve Velké válce algoritmů (zejména psaní efektivních toků). Eleanor byla vychovávána jako sirotek v režimem spravované internátní škole. Byla problémové a velmi netrpělivé dítě. Často podvědomě vymýšlela algoritmy efektivnější než bylo nutné, ale naučila se svoje názory skrývat, aby se nedostala do problémů.

S jejími 18 narozeninami byla povolána na FIT, aby pracovala na oddělení cenzury. Pro její nadání se již po roce stala seniorní cenzorkou, ovšem ona svou práci nenávidí. Dnes je den jako každý jiný, ovšem jeho průběh navždy změní její život.

Kapitola 1

Jako každý pracovní den se Eleanor v sobotu ráno vydala do práce. Přišla do své kanceláře v oddělení cenzury FIT a prošla seznamem úkolů, které bylo každé ráno potřeba udělat. Seřadila příchozí dopisy pomocí opakovaného hledání toho, který je abecedně nejmenší. Vyškrtala černou cenzorofiksou nevhodné výrazy ze zítřejších novin a předala je ke zpracování editorům.

Po třech hodinách práce Eleanor dokončila tyto triviální úkoly. Frustrovaná banalitou ranních úkolů a způsobenou ztrátou času se Eleanor pustila do čtení úkolů, které cenzorky dostaly pro dnešek na vyřešení. Dnešní seznam byl obzvláště vypečený, a tak se pomalu smířovala s tím, že úkoly nestihnou vyřešit. „Přeci se jen tak nevzdám,“ řekla si. Dopad za nesplnění plánu by postihl zbytek cenzorek, a nejspíše by některé přišly o práci. Eleanor se tiše rozhodla, že udělá všechno pro to, aby úlohy vyřešila včas.

Kapitola 2

Poté, co Eleanor udělala efektivně náročné úkoly, obává se nejhoršího, že na to přijde systém a zatknou ji, možná i popraví. Proto se rozhodne pro další logický krok, jít do baru se opít. Než si Eleanor stačila objednat, přišla za ní sličná slečna se slovy „Zdravičko, mé jméno je Claire. Smím Vás pozvat na drink? Nebojte, nejsem od

systému!“ Claire se Eleanor na první pohled líbila, navíc už neměla co ztratit, a tak se rozhodla jí věřit.

O pár panáků později si Claire všimla dvou podezřelých chlápků, kteří vešli do baru. Claire rychle popadla Eleanor za ruku a odvedla ji za roh, kde vešly do tajné chodby. Tam Claire přimáčkla Eleanor ke stěně a potichu ji zašeptala do ouška „Tě už není cesty zpět! Systém o tobě ví! Tvá jediná šance je přidat se k odboji – přidat se k nám. Tak co?“ Eleanor, rudá až po uši, jen lehce přikývla – zejména kvůli sympatiím ke Claire. Další události si už bohužel kvůli vallnému množství panáků nepamatuje. . .

Probrala se až další den ráno na pohodlné postýlce. Na židli vedle ni seděla Claire, která ji přivítala se slovy „Krásné ráno ospalko! Vítej u odboje. Představím tě vedoucímu našeho oddělení!“ Claire vzala Eleanor do vedlejší místnosti, kde byla pouze pracovní židle a stůl. Nežli se Eleanor stačila vzpamatovat, židle se otočila o 180 stupňů. Seděl na ní drobnější muž, který hladil velkého kocoura: „Vítej Eleanor. Jmenuji se Andrej Důbrý a vedu zdejší oddělení. Podle Claire jsi celkem šikovná,“ řekl s mrknutím na Eleanor, která se začervenala, „nicméně budeme si to muset trošičku prověřit! Než tě vezmeme do našich řad, měl bych tu pro tebe 5 netriviálních úkolů, které prověří tvé schopnosti.“

Úloha č. 1

Srovnání účtů



Rozmysli, popiš a naprogramuj!

10 b

Než ze sebe Eleanor stihla vypustit jakýkoliv projev souhlasu či nesouhlasu, Andrej se rozmluvil o jejím prvním úkolu. „Dostali jsme informaci, že systém se pokusí detekovat a zmrazit naše bankovní účty. To jim samozřejmě nemůžeme dovolit, neboť by to znamenalo náš konec. Potřebujeme tak účty co nejlépe a za použití co nejméně prostředků zakamuflovat – a to bude tvá první zkouška.“ Eleanor pouze tiše polkla a pokývala hlavou s tím, že se tento úkol pokusí splnit. Vnitřně se jí však zhostila nervozita, jelikož hned ze začátku s úkolem takovéto důležitosti nepočítala.

Andrej společně s Claire poté začali vysvětlovat v čem tkví podstata problému. Systém se bude pokoušet nalézt účty odboje tak, že ve všech bankách důkladně prověří vlastníky účtů s nejvyššími zůstatky kreditů. Aby odboj zůstal tzv. „pod radarem“, je zapotřebí jeho účty srovnat tak, aby nejvyšší zůstatek přes všechny účty odboje byl co nejnižší. To ovšem v praxi není tak jednoduché, neboť banky si účtují různé poplatky za příchozí a odchozí platby a finance odboje pro tuto operaci kamoufláže jsou značně omezené.

Konkrétně pro tento úkol tedy Eleanor dostala podrobný seznam všech účtů odboje, ve kterém je pro každý účet uvedený zůstatek, cena za odchozí platbu z daného účtu a cena za příchozí platbu na daný účet. K tomu obdržela maximální částku, kterou lze spotřebovat na přesuny kreditů mezi účty. Tato částka je nezávislá od prostředků na účtech a lze jí použít pouze na platbu poplatků za odchozí/příchozí platby. Jedinou povolenou operací je tedy přesun kreditů z jednoho účtu na jiný účet, kde cena za přesun jednoho kreditu je rovna součtu ceny odchozí platby ze zdrojového účtu a ceny příchozí platby na cílový účet. „Jaký nejnižší možný nejvyšší zůstatek na našich účtech dokážeš za tuto částku vytvořit?“ zeptal se Andrej a napsal částku na roh papíru, který ležel na stole. „Hned to bude,“ odpověděla Eleanor a pustila se do řešení prvního úkolu.

Vstup

Na prvním řádku budou dvě celá čísla N a K ($1 \leq N \leq 10^6$, $1 \leq K \leq 10^9$), kde N značí počet účtů odboje a K je maximální částka, kterou lze použít na přesuny kreditů mezi účty. Následují tři řádky, každý o N celých číslech, reprezentující seznam účtů odboje. První z těchto řádků obsahuje čísla Z_1, Z_2, \dots, Z_N , kde číslo Z_i značí zůstatek na i -tém účtu. Podobně druhý řádek z těchto řádků obsahuje čísla A_1, A_2, \dots, A_N a třetí z těchto řádků obsahuje čísla B_1, B_2, \dots, B_N . Číslo A_i značí cenu za odchozí platbu z i -tého účtu a číslo B_i značí cenu za příchozí platbu na i -tý

účet (cena za přesun X kreditů z účtu i na účet j je tedy $X \cdot (A_i + B_j)$). Pro tato čísla platí $1 \leq Z_i, A_i, B_i \leq 10^9$.

Výstup

Výstupem je jediné číslo – nejnižší možný nejvyšší zůstatek ze všech účtů, který lze získat přesouváním prostředků za celkovou částku ne vyšší než K .

Ukázkové vstupy

Vstup

```
3 7
6 0 0
1 1 1
1 1 1
```

Výstup

```
3
```

Vstup

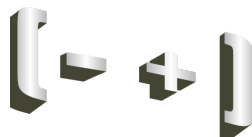
```
3 10
3 4 2
5 5 5
3 3 6
```

Výstup

```
4
```

Úloha č. 2

Ksurb



Odpověz Sfinze!

10 b

*Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.*

Jen co Eleanor úspěšně dokončila předchozí úkol, hned na ni čekal další.

„Můj synovec hraje s kamarády hru jménem Ksurb“, povídal Andrej. Eleanor hodila nervózní pohled po Claire, protože o této hře ještě neslyšela. Tu však zrovna zaujal odložený Manga komix, takže si ji ani nevšimla. Andrejovi však Eleanorin pohled neunikl, a tak se dal do vysvětlování pravidel.

„Kapitán zadá několik čísel a výsledné číslo X . Ostatní mají za úkol ze zadaných čísel vytvořit matematický výraz, jehož hodnota je právě X . Pravidla pro vytváření matematického výrazu jsou následující:“

- můžeš zvolit libovolné pořadí čísel,
- můžeš používat znaménka $+$ a $-$,
- můžeš použít libovolný počet závorek,
- mezi dvěma členy musí být znaménko.

Např. ' $3 + (2 - 1)$ ' je validní výraz, ale ' $3 (2 - 1)$ ' nebo ' $3 + - 2 1$ ' není.

Eleanor pokývala hlavou, že pravidla chápe, a přemýšlela, co asi bude jejím úkolem. „Jak vidíš“, pokračoval Andrej, „hra je celkem jednoduchá. Navíc, protože jsou kluci teprve v první třídě, používají pouze čísla do dvacítky. Bežně zvládají danou úlohu pro pět zadaných čísel, ti nejlepší dokonce i pro patnáct. Dokážeš to ty i pro víc než stovku?“

Vstup

Na prvním řádku se nachází číslo T udávající počet vstupů. Následuje T řádků, přičemž na každém řádku se nachází jeden vstup, čili jedno zadání hry.

Každý řádek začíná dvěma čísly. První je výsledné číslo X ($1 \leq X \leq 20000$) a druhé je N udávající počet čísel ve výrazu. Následuje N čísel ($1 \leq C \leq 20$), které je potřeba ve výrazu použít. Všechna čísla jsou oddělena mezerou.

Počet zadaných čísel N závisí na obtížnosti úlohy:

- *Lehčí varianta (4b)* – $1 \leq N \leq 5$
- *Střední varianta (3b)* – $1 \leq N \leq 15$
- *Těžší varianta (3b)* – $1 \leq N \leq 105$

Z úlohy lze získat částečné body, stačí pro těžší vstupy vypsát prázdný řádek.

Výstup

Pro každý vstup vypiš na samostatný řádek LZE, pokud lze sestavit matematický výraz, jehož hodnota je výsledné číslo X . Pokud žádný takový výraz sestavit nelze, vypiš NELZE.

Ukázkové vstupy

Vstup

```
4
1 3 10 5 14
8 4 4 7 5 2
8 5 5 10 15 5 20
37 8 7 4 14 17 2 16 9 4
```

Výstup

```
LZE
LZE
NELZE
LZE
```

Vstup

```
4
10000 4 20 7 8 6
4 4 1 1 1 2
3 4 1 1 1 2
4 4 1 1 2 2
```

Výstup

```
NELZE
NELZE
LZE
LZE
```

Pro výše uvedené vstupy mohly vzniknout například tyto výrazy:

$$1 = 10 + 5 - 14$$

$$8 = 4 + 2 + (7 - 5)$$

$$37 = 7 + 9 - (14 + (-(17 + 16) - 2)) + 4 - 4$$

$$3 = 1 + 1 + (2 - 1)$$

$$4 = -1 + 1 + 2 + 2$$

Úloha č. 3

Meziměstská doprava



Zamysli se!

10 b

Tato úloha je čistě teoretická, tvým úkolem zde není napsat program. Namísto toho si dej záležet na kvalitním slovním popisu, kde mimo jiné jasně zdůvodníš, proč tvůj postup skutečně bude fungovat.

Andrej pozorně sledoval Eleanorinu práci. Vedla si velmi dobře. Před zadáním dalšího úkolu se dlouze zahleděl z okna. Po dlouhé odmlce konečně promluvil: „Výborně, výborně. To se zrovna hodí. Už vím, co bude tvým dalším úkolem. Splněním dalšího zkušebního úkolu zároveň pomůžeš našemu odboji. Získávání nových členů odboje je velice riskantní, ovšem pro odboj životně nutná činnost. Je však velice obtížné najít ty správné lidi. Musí být totiž schopni pracovat efektivně a musí to být odpůrci režimu. To jsou dvě věci, kterými se v této zemi skutečně nikdo veřejně nechlubí. Jediný špatně zvolený člověk by mohl znamenat katastrofu pro celý odboj. Je proto potřeba postupovat opatrně a zjišťovat si o lidech informace.“ Chvilí zvažoval svá další slova a poté pokračoval ve vysvětlování.

„Další z tvých zkušebních úkolů je získat informace o potenciálním zájemci o členství v odboji. Budeš muset analyzovat dopravní záznamy, které od režimu ukradli členové odboje. Najdi v nich informace o této osobě, aby ji mohl jiný člen odboje dále prověřit.“

„Jak jistě víš, každý člověk má svoje unikátní ID, což je číslo v rozmezí $1 - 10^{25}$. Toto ID je ovšem unikátní pouze v rámci jednoho města. Lidé z různých měst tedy mohou mít shodné ID. Je zákonem velmi přísně zakázáno pobývat v noci v jiném městě, než v tom, ve kterém má daná osoba své bydliště.“

Dále Eleanor vysvětlil, že mezi všemi 25 městy jezdí speciální osobní přepravníky. Ty jsou jediným způsobem, jak se mezi městy pohybovat. Proto se jich také denně vypraví neskutečné množství. Města jsou číslována čísly 0 – 24. Protože si

vláda chce udržovat přehled o pohybu osob v zemi, vede si záznamy o každé takové cestě dopravníku. Aby byly záznamy o přepravě co nejkratší, zavedla následující pravidla:

- V každém dopravníku cestují pouze osoby s po sobě jdoucími ID.
- Každé ID je v dopravníku maximálně jednou a žádné místo není volné.

Díky tomu lze zápis o cestě zapsat jednoduše jako $M_1 M_2 ID_{min} ID_{max}$, kde M_1 je číslo výchozího města, M_2 číslo cílového města, ID_{min} ID osoby s nejnižším ID a ID_{max} je ID osoby s nejvyšším ID. Záznam '3 7 10 15' tedy znamená, že přepravník přepravil z města č. 3 do města č. 7 celkem 6 osob s ID 10, 11, 12, 13, 14 a 15. Bohužel z jednoho záznamu není zřejmé, z jakých měst které osoby jsou. Pokud to jde, mají úředníci občas ve zvyku slučovat záznamy více současně vypravených přepravníků do jednoho. To má za následek to, že jeden záznam o cestě může obsahovat ohromné množství osob. Na podobě záznamů to nic nemění, jen se nemůžeš spolehnout na to, že v jednom záznamu bude jen pár desítek lidí.

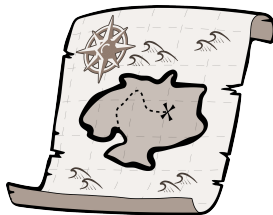
„Odboj dostal informaci o tom, že zákon o přespávání ve svém městě nějaký odvážlivec z města 0 občas porušuje. Odboj ho musí najít dříve, než ho najde vláda!“, zdůraznil Andrej.

Máš vždy k dispozici záznamy z celého dne seřazené od rána do večera. Víš, že na začátku dne začíná každý občan ve svém městě. Popiš program, který ze záznamů co nejefektivněji a nejjednodušeji zjistí, jestli jsou všichni občané večer ve svém městě. Pokud je odvážlivec z města 0 jinde, než má být, zjistí jeho ID a město, ve kterém se na konci dne nachází. Vzhledem k pečlivosti státních sil se můžeš spolehnout na validitu záznamů. Dále je naprosto nemyslitelné, aby tento zákon porušil i někdo další.

Protože je přístup k nejmodernějším technologiím velice problematický a odboj na sebe nechce upoutávat pozornost příliš vysokým odběrem elektřiny, používá pro své výpočty zařízení výkonem a parametry srovnatelné s běžnými zařízeními z roku 2018.

Úloha č. 4

Hlídky



Rozmysli, popiš a naprogramuj!

10 b

Eleanor si otřela z čela krůpěje potu a s blaženým úsměvem se rozvalila do poblíž stojícího křesla. Takový pocit za svůj život ještě nezažila. Skvělý pocit z dobře vykonané práce, z efektivního řešení úlohy. Jsou algoritmy opravdu tak silnou drogou, nebo jde spíše o důsledek spánkové deprivace, ptala se sama sebe.

Ještě než se nad celou situací stačila hlouběji zamyslet, stál u ní Andrej a v ruce držel nějaké papíry. „Prokázala si, že si dokážeš poradit s dětskými hrami, pokud chceš být opravdu plnohodnotnou součástí odboje, budeš muset ukázat více,“ pravil vážně, bez jakýchkoliv stop dřívější lůdnosti.

„Mám to pro tebe další úkol, Eleanor, je to tvá šance ukázat, že Claire nemluvila do větru,“ řekl Andrej a s posměšným výrazem se na malou chvíli otočil na Claire. „Tady ti nesu mapy některých oblastí, které sestavili naši agenti v terénu,“ ztišil trochu hlas a pokračoval ve výkladu. „Na každé z těchto map jsou vyznačena města společně s počtem kontrol, které se na té které silnici mezi dvěma městy nacházejí. Tvým úkolem bude pro dvě zadaná města najít takovou cestu, aby počet kontrol na nejhlídanější silnici na trase mezi nimi byl mezi všemi možnými cestami nejmenší.“ „Ale, k čemu něco takového je? Nebylo by pro agenti lepší používat tu nejkratší trasu?“ přerušila Andreje Eleanor. „Kdo se moc ptá, moc se dozví,“ opáčil Andrej. Následně však dodal: „Kontroly na silnicích sice fungují neefektivně, ale čím více jich je, tím existuje větší riziko, že některého z našich agentů odhalí.“ Na chvíli byl na Andrejově tváři znatelný tak trochu potouchlý úsměv, ale vmžiku se otočil a byl ten tam. Eleanor se tedy dala do práce. . .

Vstup

Na prvním řádku vstupu budou čísla n, k , kde n značí počet měst na mapě a k značí počet všech cest mezi městy.

Následovat bude k řádků, přičemž na každém řádku budou tři čísla $0 \leq a \leq n$, $0 \leq b \leq n$, $1 \leq h \leq n^2$, která označují počet hlídek h na (obousměrné) silnici mezi městy a a b .

Po zadání mapy bude na samostatném řádku následovat číslo q , což je počet cest, které musí Eleanor naplánovat. Ty budou popsány na dalších q řádcích, kde každý bude obsahovat dvě čísla s a c značící dotaz na cestu z místa s do místa c .

Můžeš se spolehnout na to, že měst v mapě nebude více, než 2000, počet cest nepřekročí 100000 a počet dotazů bude maximálně 1000.

Výstup

Výstupem programu budou vždy dva řádky pro každý dotaz, tedy celkem $2q$ řádek.

Odpověď na jeden dotaz q_i se bude skládat z identifikátorů měst přes které je třeba přejít na prvním řádku, a z počtu hlídek na nejhlídanější silnici v této cestě na řádku druhém.

V případě, že mezi dvěma zadanými městy žádná cesta neexistuje, indikuj tuto skutečnost vypsáním čísla -1 na místo počtu hlídek na nejhlídanější silnici. Místo identifikátorů měst nachazejících se na cestě mezi počátečním a koncovým městem pak můžeš vpsat libovolnou posloupnost čísel.

Ukázkové vstupy

Vstup

8 12
1 8 2
1 2 4
1 3 7
8 6 3
8 7 14
8 2 1
2 4 9
2 3 5
3 4 1
4 7 8
4 5 6
5 6 2
3
1 7
3 5
1 3

Výstup

1 8 2 3 4 7
8
3 2 8 6 5
5
1 2 3
5

Úloha č. 5

Cenzoři



Odpověz Sfinze!

10 b

*Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.*

Andrej, spokojen s dosavadní Eleanořinou prací, řekl „Poslední z úkolů je následující. Je dáno pole čísel o délce N , které jsou na začátku nastaveny na 0. Úkolem je zpracovat 3 typy dotazů:

- 0 b e – vypiš minimum na intervalu od b do e ,
- 1 b e A – přičti hodnotu A všem prvkům na intervalu od b do e ,
- 2 b e A – nastav všechny prvky na intervalu b až e na hodnotu A .“

Eleanor zarazilo, že je zadání tak krátké. Hned věděla, že jedno z možných řešení spočívá v implementaci datové struktury, která pro ni bude výzvou^[1]. Zasedla ke stroji a začala psát.

Vstup

Vstup bude zadán *nestandardním* způsobem: dostaneš pouze parametry pro jeho generování. Generování vstupu bude probíhat pomocí funkce `nextInt`, která je definovaná následovně:

$$\text{nextInt} : x = ((x \cdot a + b) \bmod 1000000007).$$

Na prvním řádku vstupu je číslo udávající počet testovacích sad T (v rozsahu $1 \leq T \leq 100$), následují jednotlivá zadání testovacích sad.

Každá testovací sada obsahuje právě jeden řádek s následujícími proměnnými: t, N, a, b, x_0 , kde $1 \leq t \leq 10^6$ je počet operací, který bude proveden, $0 < N \leq 10^6$ je velikost pole čísel a $0 \leq a, b, x_0 < 10^9 + 7$ jsou parametry pro funkci `nextInt`. x_0 je hodnota x v prvním kroku.

Opravdové vstupy si pak vygeneruješ takto:

$$\begin{aligned}t &= \text{nextInt}() \bmod 3 \\b &= \text{nextInt}() \bmod N \\e &= \text{nextInt}() \bmod N\end{aligned}$$

¹ https://en.wikipedia.org/wiki/Segment_tree

$$\text{if}(b > e) : \text{swap}(b, e)$$
$$A = \text{nextInt}() \bmod N$$

Tento postup je nutné provést pro každou operaci na vstupu. Číslo A se sice pro vstup typu 0 nepoužije, ale i tak, je nutné funkci zavolat, aby se přepočítalo x . Význam výsledných čísel je následující:

- t : Druh dotazu který má být proveden
- b : Začátek intervalu
- e : Konec intervalu
- A : Přičítané/nastavované číslo

Výstup

Za každou testovací sadu vypiš tři čísla na třech řádcích:

- Na prvním řádku bude xor všech **nejnižších** ID na intervalu pro všechny dotazy typu 0.
- Na druhém řádku bude xor všech **nejvyšších** ID na intervalu pro všechny dotazy typu 0.
- Na třetím řádku bude xor všech **sum** všech ID na intervalu pro všechny dotazy typu 0.

Tj. pokud bude pro jednu testovací sadu potřeba zpracovat 5 dotazů typu 0 a pro jejich intervaly bude součet ID roven 0, 0, 3, 9 a 33, tak bude třetí řádek výstupu pro tuto testovací sadu 43, protože $\text{xor}(0, 0, 3, 9, 33) = 43$.

Pokud ti nepůjde některá z operací, můžeš místo ni tisknout 0 a můžeš tak získat parciální body.

Ukázkové vstupy

Následující dvojice vstupu a výstupu ti může přijít z testovacího systému. Vstup se dá do generátoru, a výstup je XOR opravdových odpovědí.

Vstup

```
2
3 3 4 5 5
10 10 10 5 4
```

Výstup

```
1
1
2
10
6
43
```

Následující dvojice vstupu a výstupu jsou data, se kterými budeš opravdu pracovat. Vstup vznikl za pomoci generátoru, a na výstupu jsme pro ilustraci vyznačili stav pole po každé operaci typu 1 a 2, a výstup pro každý dotaz typu 0.

Vstup

```
sady: 2
[formát] t: (b,e) [A]

1: (0,2) [1]
0: (1,2) [0]
2: (0,1) [2]

0: (5,5) [5]
2: (5,5) [5]
2: (0,0) [0]
0: (0,0) [0]
1: (3,4) [3]
0: (3,3) [3]
1: (2,9) [9]
0: (9,9) [9]
0: (1,4) [3]
1: (4,4) [4]
```

Výstup

```
pole: 0 0 0
výstup: 1 1 1
výstup: 1 1 2
pole: 2 2 1
pole: 0 0 0 0 0 0 0 0 0 0
výstup: 0 0 0
pole: 0 0 0 0 0 5 0 0 0 0
pole: 0 0 0 0 0 5 0 0 0 0
výstup: 0 0 0
pole: 0 0 0 3 3 5 0 0 0 0
výstup: 3 3 3
pole: 0 0 9 12 12 14 9 9 9 9
výstup: 9 9 9
výstup: 0 12 33
pole: 0 0 9 12 16 14 9 9 9 9
```