72. ročník Matematické olympiády – 2022/2023

Úlohy krajského kola kategorie P

Krajské kolo 72. ročníku MO kategorie P se koná v úterý 17. 1. 2023 v dopoledních hodinách. Na řešení úloh máte 4 hodiny čistého času. V krajském kole MO-P se neřeší žádná praktická úloha, pro zajištění rovných podmínek řešitelů ve všech krajích je použití počítačů při soutěži zakázáno (s výjimkou nahrávání řešení do odevzdávacího systému). Zakázány jsou rovněž jakékoliv další pomůcky kromě psacích potřeb (např. knihy, výpisy programů, kalkulačky, mobilní telefony).

Řešení každé úlohy vypracujte na samostatný list papíru. V případě distanční organizace krajského kola poté řešení oskenujte či nafoťte a nejpozději do 20 minut po skončení soutěže nahrajte ve formátu PDF do odevzdávacího systému, který najdete na *https://osmo.matematickaolympiada.cz/*.

Řešení každé úlohy má obsahovat:

- *Popis řešení,* to znamená slovní popis principu zvoleného algoritmu, *argumenty zdůvodňující jeho správnost* (případně důkaz správnosti algoritmu), *diskusi o efektivitě* vašeho řešení (časová a paměťová složitost; to se netýká úlohy P-II-4). Slovní popis řešení musí být jasný a srozumitelný i bez nahlédnutí do samotného zápisu algoritmu (do programu). Není možné odkazovat se na vaše řešení úloh domácího kola, opravovatelé je nemusí mít k dispozici.
- Doporučujeme uvést *zápis algoritmu* v nějakém dostatečně srozumitelném pseudokódu (případně v programovacím jazyce C/C++, Python nebo podobném). Nemusíte detailně popisovat jednoduché operace jako vstupy, výstupy, implementaci jednoduchých matematických vztahů, vyhledávání v poli, třídění apod. Zápis algoritmu sice není povinnou součástí řešení, ale při nejasností v popisu řešení může opravovatelům pomoci s pochopením algoritmu.

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

Vzorová řešení úloh naleznete krátce po soutěži na webových stránkách olympiády *https://mo.mff.cuni.cz/p/*. Na stejném místě bude zveřejněn i seznam úspěšných řešitelů krajského kola a seznam řešitelů postupujících do ústředního kola. Svá opravená řešení s komentáři opravovatelů najdete v OSMO.

P-II-1 Investiční guru

Lukáš se navenek tváří jako investiční guru, který umí dokonale předvídat trh a všechny jeho investice se pouze zhodnocují. Ve skutečnosti však vlastní pouze jeden bajtcoin a každý den si zapisuje jeho aktuální hodnotu, která kolísá nahoru a dolů.

Umíte si asi představit, že když Lukáš své zápisky omylem zveřejnil, zdvihla se vlna pochybností o jeho schopnostech. Lukáš naštěstí vymyslel, jak celou situaci zachránit. Prohlásí, že si nezapisuje hodnoty pouze jedné investice, ale všech svých k investic dohromady – vždy, když se podívá na hodnotu některé z nich, tak ji zapíše hned za předchozí zapsanou hodnotu.

Pomůžete Lukášovi zjistit, jaké nejmenší k musí udat, aby bylo možné, že hodnoty všech jeho k investic pouze rostly? Nezapomeňte na důkaz správnosti svého algoritmu.

Soutěžní úloha

Na vstupu dostanete posloupnost různých přirozených čísel $a_1, \ldots, a_n$, totiž hodnoty, které si Lukáš zapsal (v tomto pořadí). Vaším úkolem je nalézt k rostoucích posloupností $(b_{1,1}, \ldots, b_{1,n_1})$, $(b_{2,1}, \ldots, b_{2,n_2})$, …, $(b_{k,1}, \ldots, b_{k,n_k})$ takových, že k je nejmenší možné a posloupnost $a_1, \ldots, a_n$ lze z těchto posloupností složit tak, že je *sloučíme*, tj. napíšeme všechny dohromady tak, aby pro každé $1 \leq m \leq k$ a pro každé $1 \leq i < j \leq n_m$ bylo $b_{m,i}$ napsané dříve než $b_{m,j}$.

Formát vstupu

Na prvním řádku dostanete jedno přirozené číslo n. Na druhém řádku dostanete n mezerou oddělených přirozených čísel $a_1, \ldots, a_n$. Můžete předpokládat, že v posloupnosti nejsou žádná dvě čísla stejná.

Formát výstupu

Na první řádek výstupu napište k, totiž nejmenší možný počet posloupností, jejichž sloučením lze získat $a_1, \ldots, a_n$. Na každém z následujících k řádků vypište mezerou oddělené členy jedné z těchto posloupností.

Příklady

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

V tomto případě je již původní posloupnost rostoucí.

Vstup:	*Výstup:*
4	2
1 3 2 4	1 2
	3 4

Další možnost by byly také například posloupnosti $(1, 3)$ *a* $(2, 4)$, *ale také třeba* $(1, 3, 4)$ *a* (2).

Vstup:	*Výstup:*
`4`	`4`
`4 3 2 1`	`4`
	`3`
	`2`
	`1`

Zde udělat nic lepšího než vyrobit čtyři jednoprvkové posloupnosti.

Bodování

Vždy bude platit $1 \leq n \leq 10^6$ a $1 \leq a_i \leq 10^9$ pro všechna $1 \leq i \leq n$. Také můžete předpokládat, že se žádná čísla v posloupnosti neopakují, tj. $a_i \neq a_j$ kdykoliv $i \neq j$.

- Až 8 bodů můžete získat za řešení, které pouze vypíše nejmenší k, ale nenajde příslušný rozklad.
- Až 6 bodů můžete získat za řešení, která předpokládají, že $k \leq 5$ (tj. vyřeší ty vstupy, v nichž lze posloupnost získat sloučením nejvýše pěti rostoucích posloupností).
- Až 3 body můžete získat za řešení, kde $n \leq 7$.

P-II-2 Developerský projekt

Studenti jedné nejmenované fakulty v jednom nejmenovaném velkoměstě si stěžovali, že budovy, kde mají výuku, jsou roztahané po celém městě. Při přejíždění mezi přednáškami tedy stráví klidně i hodinu na cestě. Děkan se rozhodl tento problém vyřešit postavením dvou nových budov poblíž stanic metra, které budou blízko sebe. Někteří členové vedení si tím ale nebyli jisti a měli pocit, že přejíždění se již stalo nedílnou součástí studia na jejich fakultě, a pokud by studenty o přejíždění připravili, mohlo by studium ztratit svou duši. Nakonec došlo ke kompromisu – budovy se postaví na dvou stanicích metra vzdálených od sebe právě k stanic. Poradíte vedení, kolik je možností, kde dvě nové budovy postavit?

Soutěžní úloha

Plánek metra si můžeme představit tak, že máme n stanic očíslovaných $1, \ldots, n$. Dvojice stanic je sousední, pokud je sousední na některé z linek metra. Navíc platí, že pokud si sousednost představíme jako hrany, pak graf stanic tvoří strom. Jinak řečeno, mezi libovolnou dvojicí stanic u, v, existuje právě jedna cesta, tedy posloupnost různých stanic $u = x_1, x_2, \ldots, x_m = v$ taková, že pro každé $1 \leq i < m$ jsou stanice x_i a x_{i+1} spojeny hranou.

Vaším úkolem je pro zadané k vypsat počet dvojic stanic takových, že se na cestě mezi nimi nachází právě k dalších stanic. Dvojice počítejte neuspořádaně, tj. u, v i v, u je ta samá dvojice, tedy ji započítejte jen jednou.

Formát vstupu

Na prvním řádku dostanete dvě mezerou oddělená přirozená čísla n a k. Na každém z dalších $n - 1$ řádků dostanete dvě mezerou oddělená přirozená čísla u, v taková, že dvojice stanic u a v je sousední. Můžete předpokládat, že každou sousední dvojici dostanete právě jednou.

Formát výstupu

Na jediný řádek výstupu vypište jedno přirozené číslo, totiž počet dvojic stanic, na cestě mezi nimiž se nachází právě k dalších stanic.

Příklady

Vstup:	*Výstup:*

`4 1`
`1 2`
`1 3`
`1 4`
	`3`

Síť metra tvoří strom, kdy všechny vrcholy jsou připojeny na vrchol 1. Tedy ve vrcholu 1 určitě nemůžeme žádnou budovu postavit – do žádného vrcholu nevede cesta přes nějaký jiný. Naopak pro libovolnou jinou dvojici vede cesta přes vrchol 1, tedy se jedná o validní řešení. Celkově tedy můžeme budovy umístit do $(2, 3)$, $(2, 4)$ *a* $(3, 4)$ *a správná odpověď je tak 3.*

Vstup:
```
4 1
1 2
2 3
3 4
```
Výstup:
```
2
```

Tentokrát síť metra tvoří cestu $1 - 2 - 3 - 4$. *Vyhovující dvojice jsou pak* $(1, 3)$ *a* $(2, 4)$.

Bodování

Ve všech vstupech platí $2 \leq n$ a $0 \leq k \leq n$. Následující seznam popisuje, kolik bodů můžete získat, budete-li předpokládat navíc následující horní odhady:

- Plný počet bodů můžete získat za řešení, které je efektivní pro $n \leq 200\,000$ a libovolné k.
- Ještě stále plný počet bodů můžete získat i za řešení, které je efektivní pro $n \leq 200\,000$ za dodatečného předpokladu $k \leq 100$.
- Nejvýš 8 bodů mohou získat řešení, které jsou efektivní pro $n \leq 200\,000$ a $k \leq 100$ za dodatečného předpokladu, že každá zastávka přímo sousedí s nejvýše pěti jinými.
- Až 5 bodů získáte za řešení efektivní pro $n \leq 10\,000$.
- Až 3 body dostanete za řešení efektivní pro $n \leq 400$.

P-II-3 Brusinková čokoláda

Dan se skupinou přátel vyhrál brusinkovou čokoládu. Každý z nich se už moc těší, až ochutná brusinku, ovšem zjistili, že i když se jedná o brusinkovou čokoládu, tak brusinek je tam poměrně málo. Přesněji řečeno, na každého z nich vychází právě jedna brusinka.

Dohodli se tedy, že rozdělí čokoládu tak, aby každý z nich dostal obdélníček čokolády obsahující brusinkový dílek (a potenciálně nějaké další dílky).

Rozdělit čokoládu mezi přátele dostal za úkol Dan. Ovšem Dan má čokoládu i brusinky opravdu rád, a tak si řekl, že by rád obojího získal co nejvíce. Zjistil, který dílek obsahuje největší brusinku, a určitě chce tento dílek. Dále by rád získal co nejvíce dílků čokolády.

Soutěžní úloha

Tabulka čokolády má r řádků a s sloupců (tedy celkem $r \times s$ dílků). Je-li n přátel, pak v právě n dílcích je brusinka. Vaším úkolem je rozdělit čokoládu na obdélníky tak, aby každý obdélník obsahoval právě jeden dílek s brusinkou, Danův obdélník byl co největší (obsahoval co nejvíce dílků) a každý dílek byl v právě jednom obdélníku.

Formát vstupu

Na prvním řádku dostanete dvě čísla r a s – rozměry čokolády (tedy počet řádků a sloupců čokolády). Na následujícím řádku je číslo n značící počet přátel (včetně Dana), a tedy i počet brusinkových dílků čokolády. Na každém z následujících n řádků je popsaný jeden brusinkový dílek pomocí dvojice čísel x_i a y_i značící, v kolikátém řádku a sloupci se daný dílek nachází. Řádky a sloupce indexujeme od jedničky, tedy platí, že $1 \le x_i \le r$ a $1 \le y_i \le s$. První z uvedených dílků obsahuje největší brusinku, takže jej chce Dan pro sebe.

Formát výstupu

Pokud nelze vhodným způsobem čokoládu rozlámat, vypište `NELZE`. V opačném případě vypište n řádků. Každý bude popisovat jeden obdélník pomocí čtveřice čísel a_i, b_i, c_i a d_i – souřadnice levého horního dílku a pravého spodního. Musí tedy platit, že $1 \le a_i \le c_i \le r$ a $1 \le b_i \le d_i \le n$. Obdélník pak obsahuje všechny dílky na souřadnicích x_i, y_i, pro než platí $a_i \le x_i \le c_i$ a $b_i \le y_i \le d_i$. Na pořadí obdélníků na výstupu nezáleží.

Musí platit, že každá brusinka je obsažena v právě jednom obdélníku a každý obdélník obsahuje právě jednu brusinku. Obdélníky se nesmí překrývat a dohromady musí pokrýt celou čokoládu.

Mezi rozděleními na obdélníky splňujícími tuto podmínku vypište takové, ve kterém má obdélník obsahující první brusinku největší obsah. Pokud je takových řešení stále více, můžete vypsat libovolné z nich.

Bodování

- Řešení za 10 bodů by mělo být schopné efektivně vyřešit úlohu pro $n \leq 10^6$ a $r, s \leq 10^4$.
- Až 8 bodů můžete obdržet, za předpokladu, že vaše řešení je efektivní alepsoň pro $r, s \leq 10^3$.
- Až 6 bodů můžete získat, když budete navíc předpokládat, že $r, s \leq 200$.
- Až 5 bodů můžete získat za řešení, které pracuje efektivně pro $r, s \leq 35$.
- Až 3 body můžete získat, pokud nebudete optimalizovat, aby Dan získal největší dílek.

Příklady

Vstup:	*Výstup:*

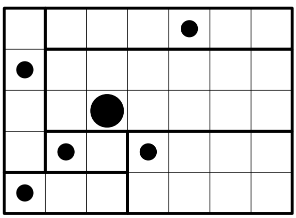
`5 7`
`1`
`3 5`
	`1 1 5 7`

Když je Dan sám, tak sní celou čokoládu.

Vstup:	*Výstup:*

`5 7`
`6`
`3 3`
`2 1`
`4 4`
`1 5`
`5 1`
`4 2`
	`2 2 3 7`
	`4 4 5 7`
	`1 1 4 1`
	`1 2 1 7`
	`5 1 5 3`
	`4 2 4 3`

Situace je vyobrazena na následujícím obrázku:



P-II-4 Hvězdní věštci

K této úloze se vztahuje studijní text uvedený na následujících stranách. Doporučujeme vám nejprve si přečíst studijní text a až potom se vrátit k samotným soutěžním úkolům. Ze stejného studijního textu budou vycházet i úlohy v dalších kolech soutěže. Studijní text je totožný se studijním textem ze zadání domácího kola.

a) *(5 bodů)*

Předpokládejme, že Hvězdné impérium tvoří cestu, tj. systémy lze seřadit do posloupnosti $s_0, s_1, ..., s_{N-1}$ tak, aby systém s_0 sousedil pouze se systémem s_1, s_{N-1} pouze se systémem s_{N-2} a pro $i = 1, ..., N-2$ systém s_i sousedil pouze se systémy s_{i-1} a s_{i+1}. V některých ze systémů probíhá povstání. Navrhněte algoritmus, který rozhodne, zda povstání probíhá v alespoň $N/3$ systémech. V zápisu algoritmu můžete použít proměnnou P, jejíž hodnota je `true`, pokud v tomto systému probíhá povstání, a `false` jinak.

b) *(5 bodů)*

Navrhněte algoritmus, který rozhodne, zda Hvězdné impérium je strom. Celková odpověď tedy má být NE pouze v případě, že Hvězdné impérium obsahuje kružnici, tj. pro nějaké $3 \leq k \leq N$ existuje posloupnost navzájem různých systémů $s_0, s_1, ..., s_{k-1}$ takových, že s_0 sousedí s s_{k-1} a s_1, s_{k-1} sousedí s s_0 a s_{k-2} a pro $i = 1, ..., k-2$ systém s_i sousedí se systémy s_{i-1} a s_{i+1} (systémy této posloupnosti mohou ale mít i další sousedy).

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

Studijní text

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

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

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

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

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

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

Příklad 1: Tři týmy

Chceme vědět, zda se hvězdné systémy mohou rozdělit do tří týmů tak, aby žádné dva systémy patřící do jednoho týmu nesousedili. Vládce každého systému si tedy nechá vyvěštit dva bity 01, 10 nebo 11 (pokud by věštec vyvěštil 00, rovnou ho popraví), udávající číslo týmu, do kterého bude jeho systém patřit. Toto číslo pak pošle všem sousedním systémům. Odpověď je NE, pokud od některého ze sousedních systémů dostane stejné číslo.

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

Popsaná strategie používá $K = 2$.

Příklad 2: Univerzální řešení

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

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

Popis algoritmu

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

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

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

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

Příklad 3: Cesta

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

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

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

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

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

pozice send[D], receive[D];

if (D > 2)
  return NE;

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

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

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

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

return ANO;
```

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

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