

Řešení úloh odevzdávejte pomocí webového rozhraní, které je dostupné na stránkách olympiády <https://mo.mff.cuni.cz/>. Tam také najdete podrobnější instrukce k odevzdávání a další informace o kategorii P.

Úlohy P-I-1 a P-I-2 jsou praktické. Vaším úkolem v nich je vytvořit a odladit efektivní program v jednom z následujících programovacích jazyků: C 11, C++ 20, Rust 2024, Python 3.13, C# 12, Java 25 nebo Pascal. Řešení těchto dvou úloh odevzdávejte přes webové rozhraní ve formě zdrojového kódu. Odevzdaná řešení budou automaticky vyhodnocena pomocí připravených vstupních dat a výsledky vyhodnocení se dozvíte krátce po odevzdání. Pokud váš program nezíská plný počet 10 bodů, můžete své řešení opravit a znovu odevzdat.

Vzhledem k různé výkonnosti stejného algoritmu implementovaného v různých programovacích jazycích nezaručujeme, že je ve všech podporovaných jazycích možné získat plný počet bodů – může se stát, že program nestihne doběhnout do časového limitu, i když implementuje algoritmus s optimální časovou složitostí. Časové limity jsou (s rezervou) nastavené dle autorských řešení implementovaných v C++.

Úlohy P-I-3 a P-I-4 jsou teoretické, za každou z nich lze získat až 10 bodů. Řešení musí obsahovat popis algoritmu, zdůvodnění jeho správnosti a v úloze P-I-3 též odhad časové a paměťové složitosti. Nemusíte psát program, algoritmus stačí zapsat ve vhodném pseudokódu nebo dokonce jenom slovně, ale v tom případě dostatečně podrobně a srozumitelně. Hodnotí se nejen správnost, ale také efektivita zvoleného postupu řešení. Řešení obou teoretických úloh odevzdávejte ve formě souboru typu PDF přes výše uvedené webové rozhraní.

V řešení teoretických úloh nemusíte detailně popisovat jednoduché operace jako vstupy, výstupy, implementaci jednoduchých matematických vztahů, vyhledávání v poli, třídění apod. Blíže viz <https://mo.mff.cuni.cz/p/knihovna.html>.

Řešení všech úloh můžete odevzdávat do **15. 11. 2026**. Opravená řešení a seznam postupujících do krajského kola najdete na webových stránkách olympiády o několik týdnů poté.

P-I-1 Skříň

Honza má rád informatiku. Má ovšem rád i chemii, a tak si ve škole vymohl vlastní klíč od laboratoře a tráví v ní všechna odpoledne u pokusů.

Chemikálii je ve škole spousta a skladují se ve sklepe, kde má každá lahvička svou vlastní uzavřenou příhrádku. Nahoru do laboratoře se nosí jen to, s čím se právě pracuje, a stojí to v jedné skříni se skleněnými dveřmi. Pro ni platí přísný předpis: dvě látky, které spolu reagují, v ní nesmějí stát zároveň. Kdyby některá lahvička netěsnila, byl by z toho v lepším případě smrad, a v horším nová laboratoř.

Které dvojice to jsou, se naštěstí pozná snadno: každá lahvička nese od dodavatele evidenční číslo (přírozené číslo větší nebo rovné dvěma) a dvě látky spolu reagují právě tehdy, když jejich evidenční čísla mají společného dělitele většího než jedna. Proč to tak vychází, netuší ani dodavatel.

Jak Honza postupuje ve svých pokusech, potřebuje pořád další a další látky. Pokaždé seběhne do sklepa, přinese lahvičku nahoru a zkusí ji postavit do skříně. Když uvnitř nestojí nic, s čím by reagovala, je vyhráno: látku má po ruce a do laboratorního deníku si zapíše ANO. Jinak ji musí odnést zase zpátky dolů, zapsat NE a ten pokus odložit na jindy.

A když s nějakou látkou skončí, odnese ji do sklepa, aby ve skříni nezabírala místo a nepřekážela ostatním. Vracení se do deníku nezapisuje. Občas se mu z roztržitosti stane, že se ze skříně pokusí odnést i látku, která tam zrovna není (třeba proto, že se ji mu tam minule nepodařilo postavit, nebo se o to dokonce nikdy ani nepokoušel). V tom případě samozřejmě nemá co odnášet a nestane se nic.

Jenže probrat před každou cestou celý obsah skříně chvíli trvá a schodů do sklepa je hodně. Tady se konečně hodí Honzova první záliba: napište mu program, který mu u každé látky rovnou řekne, jestli se do skříně vejde.

Soutěžní úloha

Skříň je na začátku prázdná. Na vstupu dostanete posloupnost q požadavků dvou typů:

- *uložení* látky x : pokud ve skříni není žádná látka y taková, že $\text{nsd}(x, y) > 1$, uložte látku x do skříně a vypište ANO, v opačném případě neprovádějte nic a vypište NE
- *vrácení* látky x : pokud je látka x ve skříni, vyjměte ji, v opačném případě neprovádějte nic; nic se nevypisuje

Výrazem $\text{nsd}(x, y)$ značíme největšího společného dělitele čísel x a y . Od každé látky má škola jen jedinou lahvičku, takže tu, která už ve skříni stojí, do ní podruhé uložit nejde. Podmínka výše na to pamatuje sama, protože pro $x \geq 2$ je $\text{nsd}(x, x) = x > 1$.

Vaše řešení by mělo číst vstupní data ze standardního vstupu a výsledky vypisovat na standardní výstup; více detailů naleznete na <https://mo.mff.cuni.cz/p/otazky-a-odpovedi.html>.

Formát vstupu

Na prvním řádku vstupu je jedno přirozené číslo q ($1 \leq q \leq 2 \cdot 10^5$) udávající počet požadavků. Každý z následujících q řádků obsahuje dvě mezerou oddělená celá čísla t a x (t je 1 nebo 2, $2 \leq x \leq 10^6$). Řádek „1 x “ znamená, že Honza zkouší uložit látku x do skříně, řádek „2 x “ znamená, že se látka x vrací do sklepa. Požadavky jsou na vstupu v pořadí, v jakém přišly.

Formát výstupu

Pro každý požadavek typu 1, v pořadí ze vstupu, vypište na samostatný řádek ANO, pokud jste látku do skříně uložili, nebo NE, pokud se musela vrátit do sklepa. Na požadavky typu 2 nevypisujte nic.

Bodování

Pro všechny vstupy platí $1 \leq q \leq 2 \cdot 10^5$ a $2 \leq x \leq 10^6$. Jsou celkem čtyři sady testovacích vstupů:

<i>body</i>	<i>podmínky</i>
2	$q \leq 1\,000$
2	$x \leq 5\,000$
2	vstup neobsahuje žádný požadavek typu 2
4	žádná další omezení

Příklady

Vstup:

6
1 6
1 10
2 6
1 10
1 15
1 7

Výstup:

ANO
NE
ANO
NE
ANO

Látku 6 uložíme do prázdné skříně. Látka 10 s ní reaguje, protože $\text{nsd}(6, 10) = 2$, a tak putuje zpátky do sklepa. Jakmile se ale vrátí i látka 6, je skřín prázdná a látku 10 už uložit můžeme. Látka 15 reaguje s látkou 10, neboť $\text{nsd}(10, 15) = 5$, zatímco látka 7 s ní nereaguje; ve skříní tedy nakonec zůstanou látky 10 a 7.

Vstup:

5
1 9
1 3
2 27
2 9
1 3

Výstup:

ANO
NE
ANO

Látky 9 a 3 spolu reagují, přestože je jedna mocninou druhé. Vrácení látky 27 neudělá nic, protože ta ve skříní nikdy nestála. Jakmile se vrátí látka 9, je skřín prázdná a látku 3 už uložit lze.

Vstup:

4

1 12

1 12

1 35

1 55

Výstup:

ANO

NE

ANO

NE

Druhý pokus o uložení látky 12 selže, protože její lahvička už ve skříni stojí a druhou škola nemá. Látka $35 = 5 \cdot 7$ s látkou $12 = 2^2 \cdot 3$ nereaguje, ale látka $55 = 5 \cdot 11$ už reaguje s látkou 35.

P-I-2 Bonsajová zahrada

Petr má bonsajovou zahradu skládající se z alejí, které jsou osázené bonsajemi, a rozcestníků, u nichž se aleje potkávají nebo končí. Protože má Petr opravdu rád stromy, tvoří i jeho zahrada strom*. Pro každé dva rozcestníky tedy existuje právě jedna posloupnost alejí, skrze které se lze dostat od prvního rozcestníku k druhému.

Některé aleje připadají Petrovi hezké a některé naopak škaredé. Petr by rád měl co možná nejhezčí zahradu, a proto si najal zahradníka. Petr nejdříve každé aleji přiřadí hodnocení v podobě celého čísla. Kladná čísla značí hezké aleje a záporná čísla škaredé aleje (alej může ohodnotit také neutrálně, tedy nulou). Čím vyšší je kladné hodnocení, tím je alej hezčí; čím nižší je záporné hodnocení, tím je škaredější. Poté zahradníkovi určí počáteční a koncový rozcestník. Zahradník následně projde od počátečního ke koncovému rozcestníku a po cestě mění všechny škaredé aleje na ekvivalentně hezké. Jelikož ale není příliš chytrý, mění zároveň všechny hezké aleje na ekvivalentně škaredé. Jinými slovy lze říct, že zahradník obrátí znaménko hodnocení všem alejím, přes které projde.

Pomůžete Petrovi zjistit, jak moc hezkou zahradu může mít?

Soutěžní úloha

Na vstupu je dán souvislý acyklický graf (tj. strom) s N vrcholy číslovanými od 1 do N a s $N - 1$ hranami, který popisuje bonsajovou zahradu. Hraný jsou ohodnocené celými čísly.

Vypočítejte největší možný součet všech hran po průchodu zahradníka. Zahradník může začínat a končit u stejného rozcestníku, v takovém případě nedojde ke změně žádné z hran.

Vaše řešení by mělo číst vstupní data ze standardního vstupu a výsledky vypisovat na standardní výstup; více detailů naleznete na https://mo.mff.cuni.cz/p/otazky_a_odpovedi.html.

Formát vstupu

Na prvním řádku je celé číslo N ($1 \leq N \leq 8 \cdot 10^4$), udávající počet vrcholů. Na i -tém z následujících $N - 1$ řádků je popis i -té hrany, který se skládá ze tří mezerou oddělených celých čísel: a_i a b_i ($1 \leq a_i, b_i \leq N$), které popisují koncové vrcholy hrany, a h_i ($-10^9 \leq h_i \leq 10^9$), které určuje hodnotu hrany. Garantujeme, že vstupní graf je strom.

Formát výstupu

Vypište jediné celé číslo, a to nejvyšší možný součet všech hran.

* Pokud nevíte, co pojem strom znamená, doporučujeme nastudovat si <https://ksp.mff.cuni.cz/encyklopedie/grafy/>.

Bodování

Pro všechny vstupy platí $1 \leq N \leq 8 \cdot 10^4$ a $-10^9 \leq h_i \leq 10^9$.

<i>body</i>	<i>podmínky</i>
1	$h_i > 0$
2	$h_i < 0$
1	$N \leq 100$
2	$N \leq 2000$
4	žádná další omezení

Příklady

Vstup:

6
1 2 3
3 1 -1
4 1 -5
5 2 -1
6 2 -5

Výstup:

5

Půjdeme z vrcholu 4 do vrcholu 6 a po cestě změníme znaménka hranám s hodnotami -5, 3 a -5.

Vstup:

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

Výstup:

17

Všechny hrany jsou kladné, a tudíž se nám nevyplatí žádné z nich měnit znaménko.

P-I-3 Skluzavky

Erik tráví léto v magickém zábavním parku. Dneska už je ale unavený, proto by se rád dostal domů, a to s vynaložením co nejméně energie. V cestě mu ovšem stojí síť magických skluzavek.

Síť skluzavek obsahuje N sloupů a mezi některými dvojicemi těchto sloupů vedou skluzavky (těch je celkem M). Erik o každé ze skluzavek ví, u kterého sloupu začíná, u kterého končí, a také jakou má délku. Mezi sloupy se dá pohybovat dvěma způsoby:

- Erik může jít po schodech podél skluzavky, a to oběma směry. Na to spotřebuje právě tolik energie, kolik je délka dané skluzavky.
- Erik se může sklouznout po libovolném počtu navazujících skluzavek. Je ovšem nutné, aby každou z použitých skluzavek projel od jejího začátku ke konci, nikoli v opačném směru. Na jedno celé sklouznutí (po libovolném počtu skluzavek) spotřebuje právě X energie.

Za zmínku stojí, že magické skluzavky v tomto parku se nezajímají o fyzikální (ani žádné jiné) zákony, jako například zákon o zachování energie. Je tedy možné, že existuje posloupnost skluzavek, po které je možné se klouzat neustále do kolečka. Výšky, v jakých se skluzavky nacházejí, jsou tedy nepodstatné, jelikož skluzavka může Erika posunout i na vyšší místo.

Formát vstupu

Na prvním řádku vstupu jsou tři přirozená čísla N , M a X – počet sloupů, počet skluzavek a množství energie nutné na jedno sklouznutí po libovolném počtu skluzavek. Erik začíná svou cestu u sloupu s číslem 1 a chce skončit u sloupu s číslem N .

Následuje M řádků popisujících jednotlivé skluzavky. Na i -tém z nich jsou tři přirozená čísla S_i , K_i a D_i udávající začátek, konec a délku i -té skluzavky (délka skluzavky je zároveň množství energie nutné na vyjití/sejití schodů podél skluzavky). Mezi každou dvojicí sloupů vede nejvýše jedna skluzavka. Síť skluzavek je souvislá, je tedy možné se nějakou cestou dostat mezi libovolnou dvojicí sloupů.

Formát výstupu

Na výstup vypište jediné číslo – minimální celkové množství energie, které Erik potřebuje, aby se dostal od sloupu s číslem 1 ke sloupu s číslem N .

Bodování

Plný počet bodů získá řešení s časovou složitostí $O((N + M) \log(N + M))$ či obdobnou, tedy efektivní pro $N, M \leq 10^5$, které najde minimální množství energie pro libovolnou síť skluzavek.

Až 7 bodů získá řešení s časovou složitostí $O(N \cdot M)$ či obdobnou, tedy efektivní pro $N, M \leq 10^4$.

Až 5 bodů získá řešení, které předpokládá, že síť klouzaček tvoří strom s nějak naorientovanými hranami. Jinak řečeno, mezi každou dvojicí sloupů existuje právě

jedna cesta (posloupnost navazujících skluzavek), kde jednotlivé skluzavky na cestě mohou být orientované libovolným směrem. Erik tedy pouze rozhoduje, kterým způsobem se bude pohybovat – trasu má jednoznačně danou.

Až 3 body získá libovolné správné řešení bez ohledu na jeho časovou složitost.

Příklady

Vstup:

5 4 100
1 2 120
2 3 40
4 3 200
4 5 90

Výstup:

390

V tomto příkladě tvoří skluzavky cestu mezi sloupy 1 a N. Erik tedy musí navštívit všechny sloupy, a to v pořadí 1 až 5. Mezi sloupy 1 a 2 se může dostat buď chůzí po schodech (energie 120), nebo sklouznutím (energie 100). Má tedy určitě lepší se sklouznout. Mezi sloupy 2 a 3 by sice bylo snazší jít po schodech (energie 40), než se sklouznout (energie 100), jelikož už se ale Erik sklouzává mezi sloupy 1 a 2, může se rovnou sklouznout ke sloupu 3 (aniž by použil další energii). Mezi sloupy 3 a 4 musí jít Erik určitě po schodech (energie 200), jelikož mezi nimi skluzavka vede v opačném směru. Mezi sloupy 4 a 5 je lepší jít také po schodech (energie 90), než se sklouznout (energie 100). Celkem tedy Erik potřebuje $100 + 200 + 90 = 390$ energie.

Vstup:

9 8 50
1 8 10
5 1 70
2 5 80
5 3 30
3 6 20
3 9 40
9 4 60
7 9 60

Výstup:

120

V tomto příkladě existuje mezi každou dvojicí sloupů právě jedna cesta. Erik nejprve vyjde schody mezi sloupy 1 a 5 (energie 70), a poté se sklouzne mezi sloupy 5 a 9 přes sloup 3 (energie 50).

Vstup:

4 5 90
4 1 80
1 2 70
2 4 60
1 3 20
4 3 50

Výstup:

70

P-I-4 Ben má nepořádek na stole

K této úloze se vztahuje studijní text uvedený na následujících stranách. Doporučujeme vám nejprve si ho přečíst, a až potom se vrátit k samotným soutěžním úkolům. Ze stejného studijního textu budou vycházet i úlohy v dalších kolech soutěže.

a) (1 bod) Konstrukce hradla NOT

Dostanete právě jeden vstupní emitör a právě jeden emitör, který bude vždy aktivní. Vstupní emitör může, ale nemusí, v čase 0 vystřelit (pokud se tak stane, nazýváme ho aktivní). Kromě toho dostanete ještě kolektor, který je s emitöry ve volné konfiguraci. Sestrojte takovou síť, aby kolektor zaznamenal paprsek právě tehdy, když vstupní emitör nebyl aktivní. Kromě výše zmíněných komponent smíte využít **pouze** rozdělovače a zrcadla, nic jiného.

b) (2 body) Konstrukce hradla OR

Dostanete právě dva vstupní emitöry, právě jeden jistě aktivní emitör a právě jeden kolektor ve volné konfiguraci. V čase 0 některé z emitörů vystřelí paprsek (nazýváme je aktivními). Sestrojte takovou síť, aby kolektor zaznamenal paprsek právě tehdy, když alespoň jeden vstupní emitör byl aktivní. Kromě výše zmíněných komponent smíte využít **pouze** rozdělovače a zrcadla, nic jiného. Navíc v této úloze musí mít kolektor právě jeden, označený, vstup.

c) (3 body) Rytmická blikání

Je dán parametr N a s ním N sudých čísel větších než 2: $x_1, \dots, x_N$. Dále dostanete N kolektorů v konfiguraci na přímce. Sestavte síť, ve které pro libovolný čas T dorazí do i -tého kolektoru paprsek ve všech časech $T + kx_i$, kde k je nezáporné celé číslo, a nikdy jindy.

Připomeňme, že k vytvoření paprsku si do sítě můžete přidat jistě aktivní emitör ve volné konfiguraci.

d) (4 body) Součet

Jsou dány parametry N a K (přirozená čísla), kde N představuje počet vstupních emitörů, které dostanete v konfiguraci na přímce. Následuje N parametrů, kde i -tý z nich představuje sílu i -tého emitöru (nezáporné celé číslo). Některé ze vstupních emitörů budou aktivní. Sestavte síť, která ověří, zda součet sil aktivních emitörů je menší než K . Pokud ano, pak kolektor ve volné konfiguraci musí zachytit paprsek, jinak nic.

Studijní text: Optická hradla

Benovi svítilo sluníčko na monitor, z čehož byl mrzutý. A aby toho nebylo málo, obdržel vysoký účet za elektřinu, protože jeho počítač s 2048 jádry má strašlivou spotřebu. Vtom dostal nápad: co když použije právě světlo jako výpočetní model? Matně si vzpomínal na hodiny fyziky, na zákon odrazu a interferenci vlnění. Ben neváhal a začal rozmísťovat zrcátka, která našel v šuplíku, po stole.

Po chvíli hraní se zrcátka měl už skoro postavenou první výpočetní síť. Vtom si uvědomil, že je to hrozně složité a nepraktické pro běžného uživatele. Netrvalo dlouho a zaradoval se. Když zatáhne žaluzie, tak paprsky budou vstupovat pouze z některých oken. Takto dokáže své síti zadat různé vstupy. Tímto se ale stavba sítě mnohonásobně ztížila, a tak vás Ben prosí o pomoc.

Nyní se podívejme na formálnější definici problému.

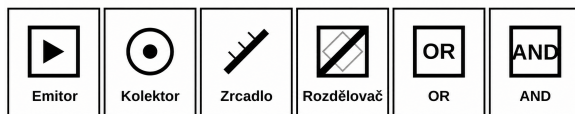
Procesor

Nacházíme se ve čtverečkové tabulce, ve které mají čtverečky velikost 1×1 . Tato tabulka je vybarvena jako šachovnice, tedy se střídají černá a bílá pole. Po těchto polích létají vodorovně či svisle paprsky.

Do tabulky můžeme umísťovat několik různých prvků. Některé druhy prvku mohou mít po stranách vstupy a výstupy, které musí být dopředu označeny. Typy prvků jsou:

- **Emitor** – poskytuje vstup pro síť. V čase 0 aktivní emitory vystřelí jedním směrem laser. To, které emitory jsou aktivní, budeme považovat za vstup řešené úlohy.
- **Kolektor** – zachycuje výstup. Pokud do něj v libovolném čase doletí paprsek, kolektor ho zaznamená. Nemá žádný výstup a může mít libovolný počet vstupů.
- **Rozdělovač** – duplikuje laser. Má označený právě jeden vstup a až tři výstupy, které jsou různé od vstupu. Po vzlétnutí laseru do vstupu dojde k rozdělení laseru a vylétnutí ze všech výstupů ve stejný moment.
- **Zrcadlo** – zahýbá laser o 90° . Přesněji určuje dvě sousední strany, pro které když paprsek přiletí z jedné strany, změní směr ve stranu druhou.
- **OR** – umožní nám získat OR paprsků, které do něj doletí ve stejný moment. Má libovolný kladný počet vstupů a právě jeden výstup, který se liší od vstupů. Paprsek vyletí z výstupu právě tehdy, když přiletí paprsek do alespoň jednoho vstupu.
- **AND** – funguje stejně jako OR, ale dělá AND – paprsek tedy vyletí právě tehdy, když ve stejný okamžik přiletí paprsky do všech vstupů.

Pokud paprsek přiletí do prvku z jiného směru, než je jeho vstup, ihned zanikne. Na každém poli může být umístěn nejvýše jeden prvek. Navíc **emitory** musejí vždy ležet **pouze na bílých polích** tabulky.



Použitá značení

Konfigurace

Rozmístění emitorů i kolektorů si ve své síti můžete zvolit, zadání úlohy však může jejich pozice nějak omezovat. V úlohách zpravidla budeme používat následující dva typy popisu možného rozmístění:

- **volná** – můžete si rozmístění a orientaci emitorů respektive kolektorů zvolit zcela libovolně.
- **na přímce** – emitory respektive kolektory musí být umístěny na vodorovné nebo svislé přímce, jejíž umístění je libovolné. Všechny rozestupy mezi emitory respektive kolektory musejí být stejné, ale přesná hodnota rozestupů je na vás. V případě emitorů v této konfiguraci všechny vystřelí stejným směrem.

Lasery

Vystřelený laser z emitoru má délku jednoho pole a letí rychlostí **1 pole za sekundu**. To, že laser prolétne skrz nějaký prvek, jako je zrcadlo nebo rozdělovač, ho nezpomalí ani nezrychlí. Paprsek letí stále rychlostí, jako by bylo pole prázdné. V případě, že se dva nebo více laserů srazí, ať už letěly proti sobě nebo kolmo, zanikají. Rovněž paprsky zanikají při nárazu do emitoru.

Parametry, vstup a výstup

Vaším úkolem je sestavit síť na základě vstupních parametrů. Přesnou podobu parametrů určuje každá úloha zvlášť, ale očekávejte formát obdobný běžným úlohám, tedy posloupnost čísel ($N, \dots$) s popsáním významem. S parametry můžete nakládat zcela libovolně, včetně libovolných předvýpočtů. Vaše řešení by mělo obsahovat obecný algoritmus konstrukce sítě, a také ideálně mnoho obrázků menších částí sítě.

Vstupem pro vaši síť bude, že některé emitory jsou **aktivní**, tedy v čase 0 vystřelí laser. Jiné jsou naopak neaktivní a nevystřelí nikdy. Toto vám už nebude dopředu známé. Pro jednoduchost značení budou emitory očíslované. Kromě těchto emitorů, které představují vstup, si můžete pro usnadnění přidat až konstantně mnoho zaručeně aktivních emitorů ve volné konfiguraci.

Výstup pak zachytí kolektory. To, pro jaký vstup by měly do jakých kolektorů doletět paprsky, specifikuje zadání každé úlohy.

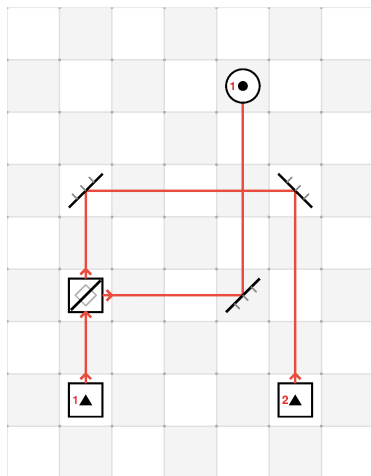
Ukázkové úlohy

Konstrukce hradla AND

Většinu prvků si umíme představit ve skutečném světě, ale problém je s hradly AND a OR. Ukažme, že hradlo AND dokážeme sestavit pouze pomocí samotného chování laseru.

Zadání: Uvažujme dva vstupní emitory, které jsou ve volné konfiguraci. Některé z nich se aktivují v čase 0. Zkonstruujte síť, v níž kolektor zachytí AND obou vstupů. Jinými slovy, kolektor zachytí laser právě tehdy, když oba lasery budou aktivní.

Řešení:



Konstrukce AND

Pohlédněme na konstrukci na obrázku výše. Rozebereme jednotlivé situace, které mohou nastat.

Pokud nevystřelí ani jeden paprsek, tak do kolektoru nemá co dorazit. V případě, že vystřelí jen emitor 1, vyruší sám sebe po rozdělení, protože paprsek letí stejně dlouho. Když naopak vystřelí pouze emitor 2, tak laser nikdy nezasáhne kolektor.

Poslední možností je, že vystřelí oba emitory. V takovém případě se vyruší paprsek emitoru 2 s jedním z paprsků emitoru 1. Druhému paprsku emitoru 1 zbyde zcela volný průchod do kolektoru.

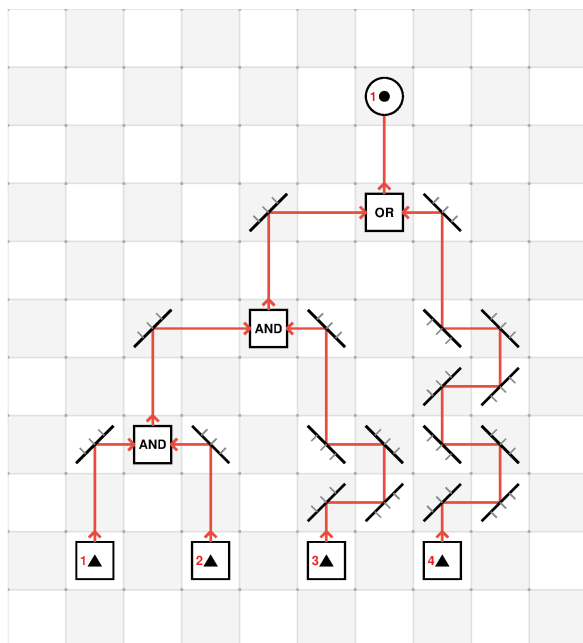
AND a OR

Ukažme, jak budou vypadat úlohy s parametry.

Zadání: Je dán parametr N . Jako další parametry dostanete sekvenci operací AND a OR délky $N - 1$. Jednotlivé operace značme $s_1, s_2, \dots, s_{N-1}$. Na vstupu sítě bude N emitorech $x_1, x_2, \dots, x_N$, které jsou v konfiguraci na přímce. Vaším úkolem je určit zápis $((x_1 \ s_1 \ x_2) \ s_2 \ x_3) \dots$.

Řešení: Jako rozestupy zvolíme 1 pole. Konstrukce celé sítě je téměř nezávislá na tom, zda v daný moment máme použít AND, nebo OR, protože se obě kompo-

nenty chovají velmi podobně. Síť stavíme postupně zleva doprava: v každém kroku přivedeme do hradla výstup z předchozí části a paprsek z dalšího emitoru. Aby se paprsky v hradle potkaly ve stejný moment, paprsek z dalšího emitoru zpozdíme pomocí záhybů (viz obrázek níže). Počet záhybů přitom s každým dalším emitorem roste o jeden (od druhého emitoru). Stejně pak postupujeme pro libovolně velké N .



Konstrukce pro úloha AND a OR, kde $N = 4$