

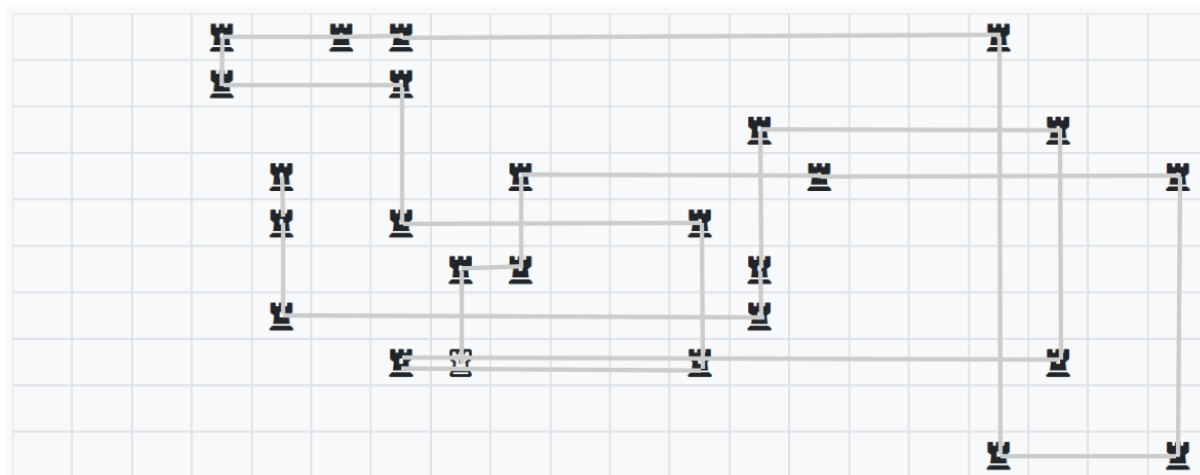


L1 Losí věže (řešení)



InterLoS 2023

Řešení úlohy je následující:



Výsledné heslo: *KLARINETISTA*



L2 Kolik vidíš losů? (řešení)



InterLoS 2023

Úloha kombinuje princip mrakodrapů a sudoku. Nejprve je nutné doplnit některá čísla podle viditelnosti. Dobré je postupovat od míst, kde je vidět jen málo losů – to nám pomůže umístit vysoká čísla. Tímto způsobem je možné doplňovat čísla od větších po menší. V pozdějších fázích už lze také využívat princip sudoku. Stále je však velmi důležité využívat viditelnosti, která nám dává (alespoň částečné) informace o uspořádání čísel.

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

Výsledné heslo: 751492368916273485627135849



L3 Losí pozemky (řešení)

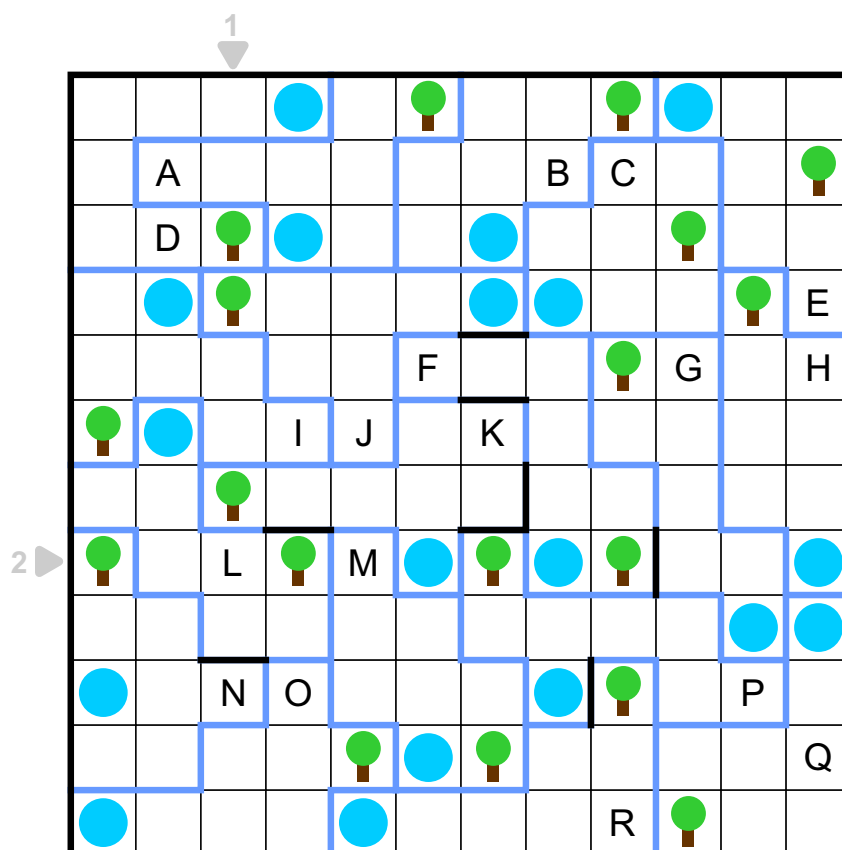


InterLoS 2023

Pro vyřešení úlohy se hodí využívat hned několika věcí:

- Která políčka určitě patří tomuto losovi? (přetékání z malých uzavřených oblastí, jiný los ho nemůže odříznout)
- Kterému losovi mohou patřit tato volná políčka?
- Ke kterým jezírkům/stromům se dostane tento los?
- Kterí losi se dostanou k tomuto jezírku/stromu?

Výsledné řešení vypadá takto:



Výsledné heslo: *DADJIIKLLNOONLLLMKPFPGGH*



L4 Pizzová (řešení)

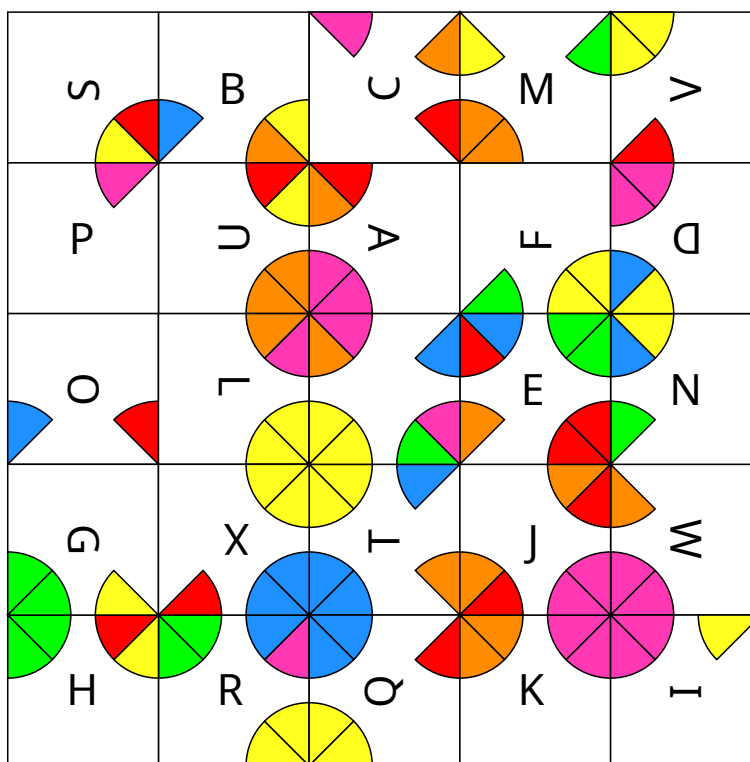


InterLoS 2023

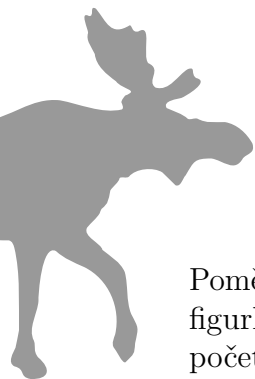
Úlohu lze řešit v následujících krocích:

1. Identifikuji, o kterou pizzu se jedná.
2. Naleznu dílky s kousky pizzy, které v dané pizzě chybí.
3. Zkusím, zda přiložením dílku nevznikne neexistující pizza nebo úsek prázdných kousků proložený neprázdným kouskem.
4. Pokud dílek je jediný splňující tato pravidla, fixně jej umístím.

Pro dané zadání mohu tyto kroky aplikovat tak, že nejdříve identifikuji, že pizza v pravém dolním rohu fixního dílku musí být 1× mořské plody, 1× salámová, 1× špenátová a 1× tuňáková. Neexistuje totiž jiná pizza, která by zároveň obsahovala dílek špenátové a tuňákové. Vyhledám možné dílky. Neexistuje dílek, který by obsahoval mořské plody i salámovou, použijí se tedy dílky samostatné. Dílky se salámovou pizzou jsou 3 (C, E a T). T nemůže být umístěn napravo od fixního dílku, protože by vzniklo proložení úseku s prázdnými dílky. Také nemůže být umístěn pod fixní dílek, protože by vznikla neexistující pizza. Podobně (akorát obráceně) to je i pro dílek C. Dílek E tedy můžeme fixně umístit napravo od fixního dílku. U dílku s mořskými plody to je obdobné. Dílky B a O nemohou být pod fixním dílkem, protože by opět vzniklo ono proložení. Jediná zbývající možnost je tedy T. Podobným způsobem lze pokračovat dál, dokud není složeno. Výsledné složení je následovné.



Výsledné heslo: *SBCMVPUAFDOLENGXTJWHRQKI*

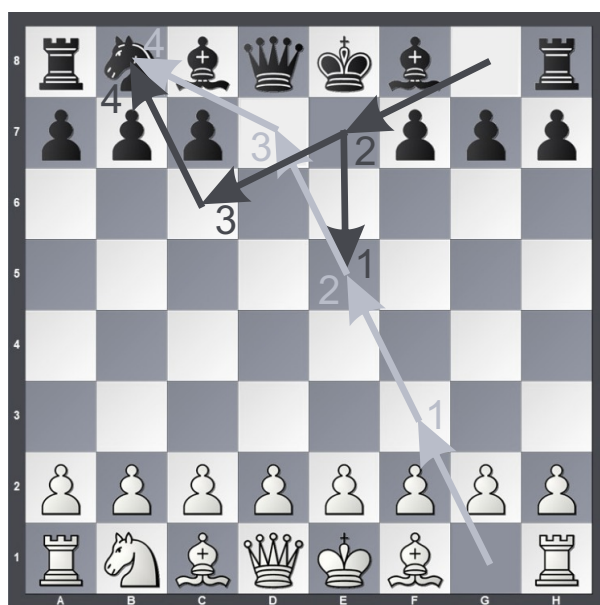


L5 Šachová (řešení)



InterLoS 2023

Poměrně snadno lze odvodit, že jsou potřeba alespoň 4 tahy (bílý musí sebrat černému 3 figurky, přitom v prvním tahu brát nemůže; poté musí černý sebrat bílého koně). Tento počet tahů je dostatečný:



Výsledné heslo: *JF3PE5JE5JE7JD7JC6JB8JB8*

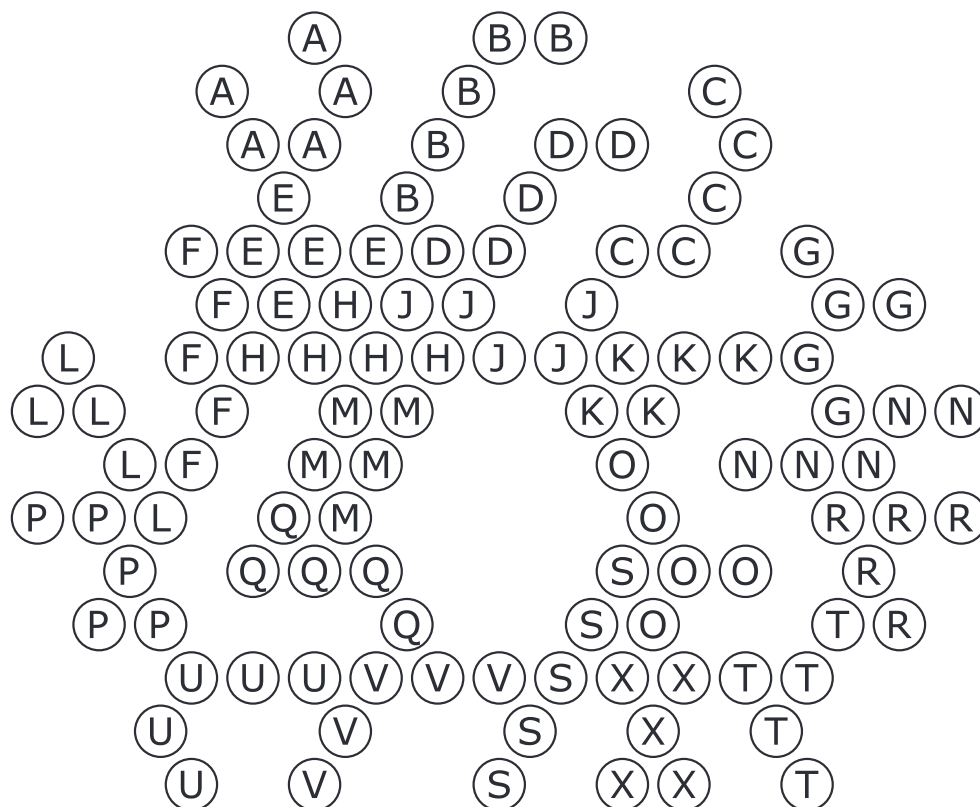


L6 Bytový dům (řešení)



InterLoS 2023

Ze začátku je jasné umístění hned několika krajních dílků. Je tedy dobré doplňovat dílky z kraje, čímž se postupně zužuje výběr.



Výsledné heslo: 234455665656555

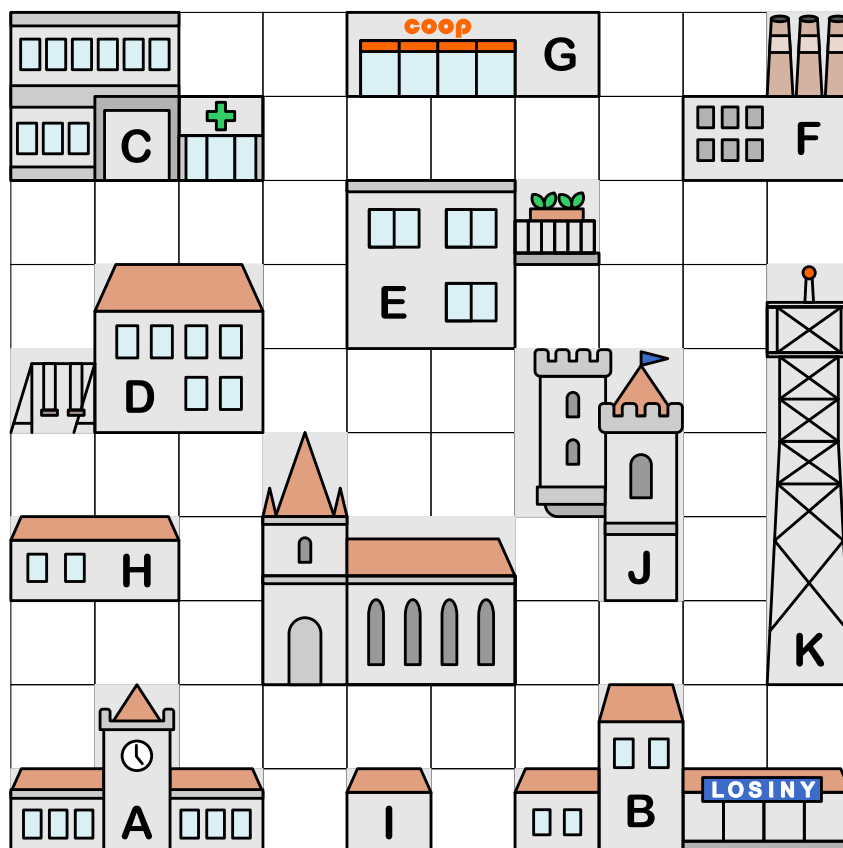


L7 Budování města (řešení)



InterLoS 2023

Je jasné, že dva nejmenší dílky musí být na levé a dolní straně kostela. Zbytek s trochou zkoušení snadno doskládáme.



Výsledné heslo: *GCFEDHJKAIB*



L8 3D Einstein (řešení)



InterLoS 2023

Dům	Zvíře			Barva			Sport		
	1	2	3	1	2	3	1	2	3
Otec	t	s	l	č	z	ž	f	h	t
Matka	k	p	ž	m	f	h	p	a	b
Dítě	d	g	o	b	o	r	l	v	s

Výsledné heslo: *TKDSPGLŽO*



L9 Světlolos



InterLoS 2023

Tato úloha byla variací na klasický rébus s žárovkami. Změnou ale bylo, že se okolo kliknutého políčka změnily žárovky pouze do tvaru T.

S každým dalším kliknutím se tento tvar otočil o 90 stupňů po směru hodinových ručiček. Tvary, kde se měnily žárovky, tedy vypadaly následovně:

```

      #      #      #
# # #  # #  # # #  # #  # # #  atd.
      #      #      #      #

```

Výsledné heslo: *KAPESNIK*



S1 Oslavná (řešení)



InterLoS 2023

Každý řádek napovídá na slovo končící na -nice. Počet písmen prvního slova v druhé větě je stejný jako počet písmen před -nice. Počet E ve slově „nice“ z druhé věty pak udává, kolikáté písmeno z tohoto slova vybereme:

HRANICE

VESNICE

STÁTNICE

LEDNICE

ROVNICE

VĚZNICE

POPELNICE

MÁRNICE

NEMOCNICE

SNĚŽNICE

UČEBNICE

VÁNICE

KOLEJNICE

BRÁNICE

SOUŘADNICE

Výsledné heslo: *ZPRONEVERA*



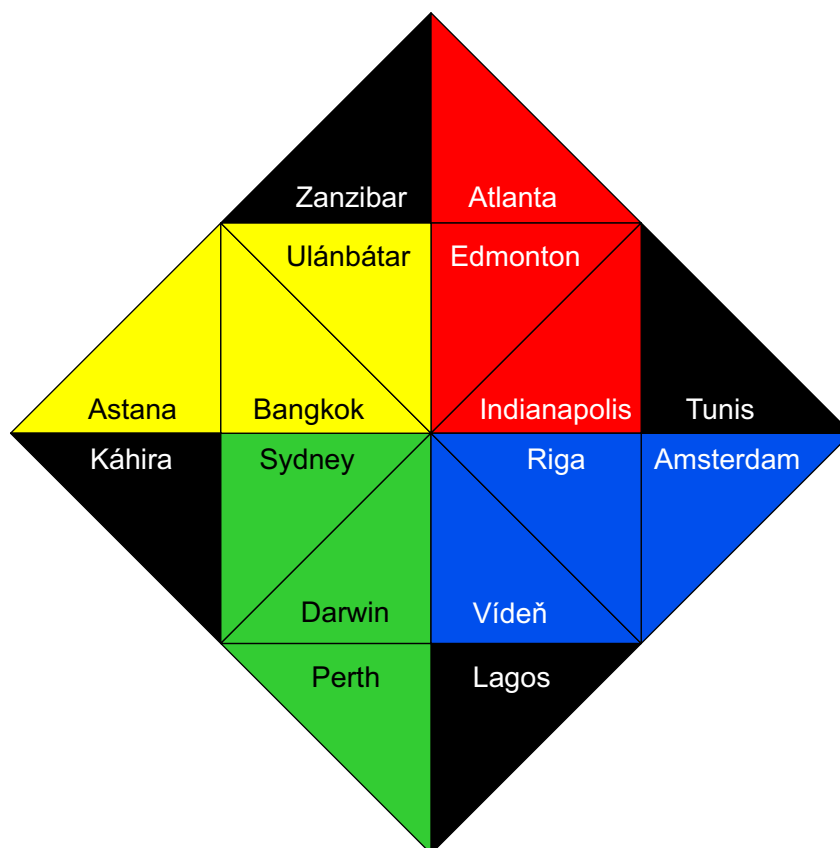
S2 Cestovní (řešení)



InterLoS 2023

Zadání nám říká, že máme najít cestu z Amsterdamu do Ulánbátaru a cestou navštívit Sydney. Jenže jakým způsobem cestu brát? Zbytek šifry připomíná hru *Nebe, peklo, ráj*. Pokud tedy čtverec vystříhneme a složíme, můžeme stejně jako ve hře *Nebe, peklo, ráj* otvírat jednotlivá políčka – a ta jsou zadána čtveřicí čtverec/kolečko, puntíky/pruhy, červená/modrá a žlutá/fialová. Nyní je praktické poznačit si, která políčka jsou čtverce/puntíky a která kolečka/pruhy, a čtverec zase rozložit.

Dále pracujeme s tím, že na každém políčku je jedna až dvě šipky s názvem města. Tyto šipky říkají, do kterých měst se lze dostat z aktuálního políčka, a udávají čtveřici, která toto město kóduje. To nám umožňuje dvě věci: doplnit jednotlivým políčkům názvy měst, a také najít nejkratší cestu z Amsterdamu do Ulánbátaru přes Sydney. Pokud přečteme první písmena měst, která po cestě mjíme, dostaneme mezitajenku *APKAZBAREVSVETADILU*. Nyní je potřeba využít i to, které políčko odpovídá kterému městu – když totiž obarvíme olympijskými barvami políčka podle toho, na kterém světadílu se dané město nachází, získáme obrázek připomínající logo Google fotek:



Výsledné heslo: *FOTKYGOOGLE*



S3 Když nevíte... (řešení)



InterLoS 2023

Nejprve se zaměříme na první část šifry. Pojmenujeme jednotlivé řádky, které odpovídají známým reklamním sloganům, na což napovídá i název šifry. Zapišeme si i značky, ke kterým reklamní slogany patří.

1. Tady je svět ještě v pořádku. [Globus]
2. Bílá bělejší, prádlo čistější. [Lanza]
3. Když máš hlad, nejsi to ty. [Snickers]
4. Zde jsem člověkem, zde nakupuji. [DM]
5. Život je hořký, bohudík. [Radegast]
6. Dlouhý život pro vaši pračku. [Calgon]
7. Když nevíte coby, najdete to v OBI. [OBI]
8. Když ji miluješ, není co řešit. [Kofola]
9. Nízké ceny pro vaše zdraví. [Dr. Max]
10. I muži mají své dny. [Fernet]

Nyní se přesuneme do pravé části šifry, kde z jednotlivých slov a písmen sloganů čteme mezitajenky. Dáváme si pozor na indexování od nuly.

- modrá: PRADLO
- červená: SALKOHOLEM
- zelená: SCUKREM
- fialová: PROZDRAVI
- oranžová: MARKETY

Nezoufáme a uvědomíme si, že orgové jsou plantážníci, kteří vymýšlejí víceurovňové šifry, a luštíme dále. Mezitajenka je 5 a sloganů 10. Navíc se každá barevná nápověda váže právě ke dvěma sloganům. Nyní přichází čas na jména produktů. Provedeme uvedené množinové operace na množinách písmen značek.

1. $\{DRMAX\} \cap \{DM\} \setminus \{M\} = \{D\}$
2. $\{LANZA\} \cap \{CALGON\} \setminus \{LN\} = \{A\}$
3. $\{SNICKERS\} \cap \{KOFOLA\} = \{K\}$
4. $\{RADEGAST\} \cap \{FERNET\} \setminus \{ET\} = \{R\}$
5. $\{OBI\} \setminus (\{GLOBUS\} \cap \{OBI\}) = \{I\}$
6. $\{DRMAX\} \cap \{DM\} \setminus \{M\} = \{D\}$
7. $\{LANZA\} \cap \{CALGON\} \setminus \{AN\} = \{L\}$
8. $\{LANZA\} \cap \{CALGON\} \setminus \{LN\} = \{A\}$

Řešením je značka, která dává křídla.

Výsledné heslo: *REDBULL*



S4 Losmisměrka⁻¹ (řešení)



InterLoS 2023

Název je na $-1.$, a po chvíli zkoumání zjistíme, že všechna slova jsou otočená, tedy celkově se zde bude pracovat s nějakým převrácením.

Proškrtnáme všechna slova a čteme odzadu: „doveď inverzi k dokonalosti“. V klasické osmisměrce se čtou ta písmena, která nejsou proškrtnutá, neboli jsou nejméně proškrtnutá, tedy inverzí budou ta, která byla přeškrtnuta nejvíce. Odzadu nám vyjde následující (číslo je počet proškrtnutí): 6 – HESLO, 5 – JE, 4 – JOKER, 3 – „“, dále už je to úplná změň.

Výsledné heslo: *JOKER*

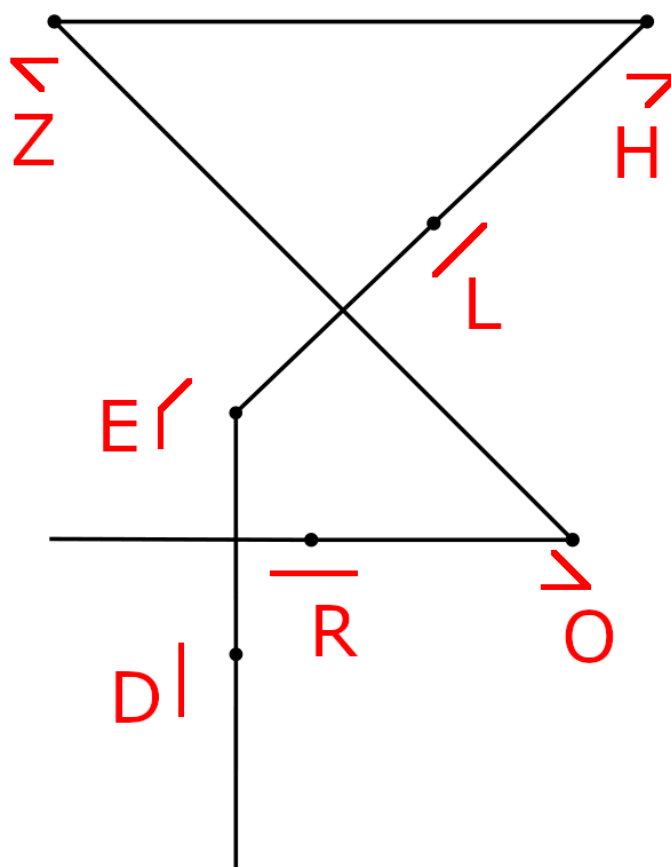


S5 Lomená čára



InterLoS 2023

Jednotlivé tečky na čáře odpovídají písmenům v semaforové abecedě.



Výsledné heslo: *ROZHLED*



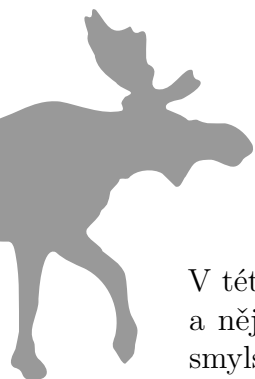
S6 Pivní (řešení)



InterLoS 2023

Každý řádek vykreslí jedno písmeno v mapě ČR, spojujeme lokace pivovarů. Vyjde ZVELKEOBCELOSU. Jako heslo zadáváme jméno piva, které se vaří ve Velkých Losinách.

Výsledné heslo: *ZLOSIN*



S7 Estetická



InterLoS 2023

V této krásné šifře jste si mohli všimnout, že na každém řádku jsou zmíněny dva smysly a nějakým slovem určené místo. Mezi obrázky byly mimo jiné i obrázky odpovídající smyslům (ruka, oko, ucho, nos).

Pokud jste šifru přeložili tak, že na slovem určeném místě byl první smysl, pak byl druhý smysl na nějakém obrázku.

Když jste vzali první písmena těchto obrázků, dostali jste výsledné heslo.

1. podívám, sem, cítím → prase → P
2. sáhnu, tohle, slyším → orel → O
3. kouknu, tady, nahmatám → strom → S
4. poslechnu, tady, dotýkám → taxi → T
5. přičichnu, sem, uvidím → Estonsko → E
6. dotknu, sem, uslyším → los → L

Výsledné heslo: *POSTEL*



S8 Logická (řešení)



InterLoS 2023

Každý řádek obsahuje právě 5 koleček, některá obsahují obrázek (na celém řádku stejný) s měnícím se pozadím. Některá pozadí k obrázku sedí, některá ne. Když si k tomu přidáme název šifry, může nás napadnout, že se jedná o hru Logik.

Pojmenujeme obrázky a spočítáme, kolikrát jsou na správném/špatném místě:

- STROM – 1x správné, 3x špatné
- HOLUB – 2x správné
- MRKEV – 1x špatné
- RUSKO – 4x špatné
- SRDCE – 2x špatné

Následně vyřešíme vzniklou logickou hádanku – Pokud je obrázek na správném místě, znamená to, že jedno z jeho písmen je na správném místě v tajence, obdobně pro obrázek na špatném místě.

Po vyřešení logické hádanky vyjde tajenka TORUS.

Výsledné heslo: *TORUS*



S9 Cink cink (řešení)



InterLoS 2023

V šifře vidíme tramvaje a směry, kterými mohou jet. Stačí použít světelné signály pro tramvaje: XOR propojených signálů (natočených podle směru tramvaje) nám dá Braillovo písmo.

$$\begin{array}{c} \circ \\ \circ \\ \bullet \end{array} \oplus \begin{array}{c} \bullet \\ \circ \\ \circ \end{array} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \text{ V}$$

$$\begin{array}{c} \bullet \\ \bullet \\ \circ \end{array} \oplus \begin{array}{c} \bullet \\ \bullet \\ \circ \end{array} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \text{ R}$$

$$\begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \oplus \begin{array}{c} \circ \\ \circ \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \text{ S}$$

$$\begin{array}{c} \bullet \\ \circ \\ \circ \end{array} \oplus \begin{array}{c} \circ \\ \circ \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \text{ T}$$

$$\begin{array}{c} \circ \\ \bullet \\ \bullet \end{array} \oplus \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \text{ V}$$

$$\begin{array}{c} \circ \\ \bullet \\ \bullet \end{array} \oplus \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} = \begin{array}{c} \circ \\ \circ \\ \circ \end{array} \text{ A}$$

Výsledné heslo: *VRSTVA*



P1 Losí plotr (řešení)



InterLoS 2023

Interpret uvedeného jazyka je vcelku přímočarý – asi nejjednodušší řešení si udržuje dvourozměrné pole reprezentující aktuální stav vykreslovaného obrazce. Obtíž při tomto přístupu může představovat neznámá výška a šířka vykreslovaného obrazce. Neznámou výšku lze vyřešit přidáním nového prázdného řádku pokaždé, když se program pokusí přistoupit mimo pole. Neznámá šířka takto snadné řešení nemá, nicméně vzhledem k informaci, že program bude řešení vykreslovat, lze odhadnout, že řádky budou mít nějakou maximální šířku, aby se výstup dobře prohlížel (a opravdu stačí šířka 80 znaků). Řešení, které dynamicky zvyšuje šířku, jak je potřeba, ovšem není výrazně komplikovanější, jen ho nebudeme prezentovat.

Jediný zádrhel při řešení je, že první program nevykreslí řešení, ale nový ASCII-art program, který je třeba nejdříve převést na obyčejný text a tento znova spustit. V parsování meziprogramu lze ale využít tři faktů, které lze odhalit důkladným prohlédnutím meziprogramu: všechna písmena jsou stejně vysoká, každé lze jednoznačně určit už z prvního řádku a mezi dvěma znaky programu je vždy prostor právě dvě mezery široký.

Vykreslení obrázku podle tohoto nového programu už dává obrázek a heslo.

Referenční implementaci si můžete prohlédnout v souboru [@ losi-plotr-solution.py](#).

Výsledné heslo: *KROKODYL*



P2 losemafory (řešení)



InterLoS 2023

Úloha nebyla nijak zákeřná, stačilo nasimulovat to, co bylo vysvětleno v zadání. Informace o křižovatkách včetně přítomnosti losů se dala udržovat třeba pomocí seznamu seznamů reprezentujícího souřadnice se seznamem informací o stavu křižovatky a přítomnosti losů po řadě na každé odbočce na každé dvojici indexů, který poté stačilo pouze vhodně aktualizovat.

Příklad řešení v jazyce Python naleznete zde: [@ losemafory-solution.py](#)

Výsledné heslo: *872*



P3 Logo IOS s.l.o. (řešení)



InterLoS 2023

Úkolem bylo pouze odsimulovat cestu loga po monitoru, program v Pythonu může vypadat například následovně: [🔗 los-logo-solution.py](#).

Obsahuje následující dvě myšlenky:

1. Protože máme logo obdélníkové, můžeme se k němu chovat, jako kdyby to byl bod, přičemž rozšíříme oblast rámu o jeho rozměr nahoru a doleva, tím si ušetříme práci při detekci odrazů (to je obsaženo již ve funkci load).

2. Tato myšlenka je samotná práce s posunem loga po monitoru, to se provádí následovně: Pro přesný výpočet se souřadnice udávají „ve zlomcích“, tedy za jeden cyklus se přidá vždy zlomek pohybu. (Protože jmenovatel je pro x stále stejný a roven velikosti y složky vektoru, není třeba ho uvádět, analogicky pro y. Stejně tak je zjevné i znaménko.) Přičemž celá část z tohoto čísla udává souřadnici políčka, v rámci kterého se právě pohybujeme, tedy je potřeba jej pro záporný počáteční vektor nastavit o 1 nižší, než je vstupní pozice. Poté se již musíme jenom správně vypořádat s odrazy.

Pro vykreslení můžete využít následujícího skriptu: [🔗 draw-logo-solution.py](#), přičemž musíte ještě odkomentovat všechny (4) řádky s yield v původním souboru.

Výsledné heslo: 1359



P4 Optical Moose Recognition



InterLoS 2023

Pro vyřešení nebylo třeba dělat něco zbytečně chytrého. Úplně dostačující bylo projít celý obrázek a pro každé místo zkusit, jestli se tam nevyskytuje los.

Vzorová implementace v Pythonu: [🔗 optical-moose-recognition-solution.py](#)

Výsledné heslo: 1097



P5 Biodiverzita (řešení)



InterLoS 2023

Všetkých možností, ako môže vyzerat tabuľka, je 2^{64} , čo je príliš veľa, je preto potrebné spočítať výsledok nejako chytrejšie.

Možností, ako môže vyzerat jeden riadok, je však iba $2^8 = 256$. Pre každú možnosť tak vieme spočítať, akú má biodiverzitu tento riadok v rámci seba samého (bez interakcie s okolím).

Postupne budeme počítat počet možností, ako získať jednotlivé možné biodiverzity pre tabuľky veľkosti $r \times 8$ pre r od 1 do 8. Tieto možnosti si navyše budeme držat rozdelené podľa toho, akým riadkom tabuľka končí. Naším cieľom je teda spočítať hodnoty $freq[r][d][R]$, ktoré značia počet možností, ako v tabuľke veľkosti $r \times 8$ s posledným riadkom R (máme 256 možností) získať biodiverzitu d .

Pre $r = 1$ je to jednoduché, vyskúšame všetky možnosti a každá končí sama sebou.

Keď teraz budeme chcieť počty možností pre $r + 1$ riadkov, vyskúšame všetky možnosti, ako vyzerá posledný riadok a ako vyzerá predposledný (t. j. akým riadkom končí o 1 menšia tabuľka). Ak existuje f spôsobov, ako v menšej tabuľke s posledným riadkom R' získať biodiverzitu d' , tak po pridaní riadku R nám narastie o f počet spôsobov, ako získať nejakú biodiverzitu d . Rôzne kombinácie posledných dvoch riadkov môžu vyústiť v rovnakú biodiverzitu d , takže začneme s počtom možností 0 a vždy pripočítavame.

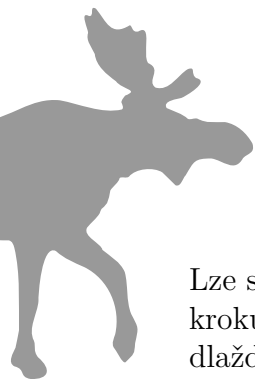
Pritom d vieme vypočítať tak, že ku d' pripočítame biodiverzitu v rámci pridaného riadku R a biodiverzitu medzi práve pridaným riadkom R a posledným riadkom zvyšku tabuľky R' . Využívame, že po pridaní riadku závisí nová biodiverzita len na poslednom riadku a jeho interakcii s posledným riadkom zvyšku.

Pre každú možnosť posledných dvoch riadkov a každú možnosť biodiverzity v menšej tabuľke spočítame, ako sa zmení biodiverzita pridaním nového riadku a aktualizujeme počet možností pre väčšiu tabuľku.

Napokon spočítame súčet možností, ako získať požadované biodiverzity 24 a 42 cez všetky možnosti posledného riadku pre tabuľku 8×8 .

Tu si môžete pozrieť implementáciu v Pythone: [📄 biodiverzita-solution.py](#).

Výsledné heslo: `105948978465860788X791201453946816`



P6 Visutý Lost (řešení)



InterLoS 2023

Lze si všimnout, že množství různých čísel je stejné jako množství řádků, tedy v každém kroku je vždy potřeba jít po mostu dopředu. To dává maximálně tři možné kroky z jedné dlaždice.

V samotném algoritmu máme 8 počátečních pozic, ze kterých hledáme cestu na jakékoliv poslední políčko. Při prohledávání si pamatujeme, na která políčka jsme už vstoupili a v jakém pořadí. A pak prohledáváme; jaký způsob volíme, je jedno, musí být ale úplný.

Výsledné heslo: *14,1,5,21,6,19,17,7,9,23,20,22,15,11,2,8,16,4,3,0,12,13,10,18*



P7 Květinový park



InterLoS 2023

Pro vyřešení této úlohy bylo třeba umět zjistit vzdálenost bodu od jiného bodu, a vzdálenost bodu od úsečky. První část je jednoduchá (pan Pythagoras poradí). Složitější je zjistit vzdálenost bodu od úsečky.

Mějme úsečku určenou dvojicí bodů $A = [x_1; y_1]$ a $B = [x_2; y_2]$ a bod $X = [x; y]$, jehož vzdálenost chceme určit.

Úsečka nám určuje přímku o vektoru $(x_2 - x_1; y_2 - y_1)$. Někde na této přímce se nachází bod T , který je nejbližší bodu X . Tento bod má souřadnice $T = [x_2 - (x_2 - x_1) \cdot t; y_2 - (y_2 - y_1) \cdot t]$. Všimněte si, že pro $0 \leq t \leq 1$ tento bod leží na úsečce.

Aby byl tento bod nejbližší bodu X musí být TX kolmé k AB , tj. skalární součin jejich vektorů je roven 0.

$$(x_2 - x_1; y_2 - y_1) \cdot (x_2 - (x_2 - x_1) \cdot t - x; y_2 - (y_2 - y_1) \cdot t - y) = 0$$

Potom už jenom stačí z rovnice vyjádřit t a dosazením do $[x_2 - (x_2 - x_1) \cdot t; y_2 - (y_2 - y_1) \cdot t]$ získáme souřadnice bodu T . Vypočítat vzdálenost od zjišťovaného bodu je zase na Pythagorově větě.

Na konec stačilo tohle celé udělat pro všechny souřadnice a všechny vrcholy a hrany chodníku.

Vzorová implementace v Pythonu: [📄 kvetinovy-park-solution.py](#)

Výsledné heslo: *50298*



P8 Popisné čísla (řešení)



InterLoS 2023

Popisné číslo, ktoré sa páči losovi Arnoštovi, musí mať zrejme dĺžku binárneho zápisu, ktorá je násobkom troch.

Ak by nás zaujímali všetky čísla dlhé $3k$ bitov, tak potrebujeme vybrať $2k$ pozícií pre jednotky a k pre nuly. Avšak prvá cifra musí byť vždy 1, takže vyberáme $2k - 1$ spomedzi $3k - 1$ pozícií a počet možností nám určuje kombinačné číslo $\binom{3k-1}{2k-1}$. Kombinačné čísla vieme vypočítať tak, že si skonštruujeme Pascalov trojuholník – tak sa vyhneme pretečeniu premenných v prípade, že by sme používali vzorec s faktoriálmi.

Číslo $M = 12345678987654321$ má 57 bitov, takže nie všetky čísla dĺžky 57 chceme započítať, ale niektoré áno. Použijeme preto takýto postup: budeme postupne konštruovať vhodné číslo a udržiavať si, koľko jednotiek ešte potrebujeme. Prvá cifra je vždy 1 a potom potrebujeme ešte $57 \cdot \frac{2}{3} - 1 = 37$ jednotiek. Postupne budeme prechádzať binárne cifry čísla M .

Ak narazíme na 0, ďalšia číslica generovaného čísla musí byť 0, inak by sme dostali väčšie číslo, ako je povolená hranica.

Ak narazíme na 1, všetky čísla majúce na tejto pozícii nulu sú určite menšie ako hranica M , ich počet teda spočítame pomocou kombinačného čísla. Spomedzi zostávajúcich pozícií (je ich toľko, koľko bitov čísla M sme ešte neprečítali) potrebujeme vybrať toľko, koľko jednotiek nám ešte chýba. Nesmieme zabudnúť overiť, že počet jednotiek je nezáporný a menší alebo rovný ako počet zostávajúcich pozícií (ak sme do čísla už napísali príliš veľa alebo príliš málo jednotiek, žiadne ďalšie vhodné kombinácie už neexistujú). Okrem toho sú ešte čísla majúce na tejto pozícii jednotku – zapíšeme si teda do generovaného čísla 1, znížime o 1 počet potrebných jednotiek a pokračujeme ďalej.

Po spracovaní posledného bitu by sme mali overiť aj to, či vygenerované číslo spĺňa Arnoštovu podmienku alebo nie – vygenerované číslo je vždy M (vždy sme zapisovali jednu jeho binárnu cifru) a naše M podmienku nespĺňa. Keby spĺňalo, museli by sme ho ešte započítať do výsledku.

Jedným prechodom cez bity čísla M tak vieme spočítať počet vyhovujúcich čísel dĺžky 57.

Všetky kombinačné čísla pre všetky dĺžky sčítame a dostaneme výsledok.

Vzorová implementácia: [@popisne-cisla-solution.py](#).

Výsledné heslo: *22976587081203*



P9 Nočný les (řešení)



InterLoS 2023

Naším cieľom je nájsť najkratšiu cestu, pričom posun na susedné políčko trvá vždy rovnako dlho, takže sa ponúka použiť prehľadávanie do šírky (BFS). Musíme si však udržiavať aktuálnu hodnotu zimy (súčet úrovní zím zatiaľ prejdejších políčok). Preto prehľadávanie nestačí spustiť na grafe tvorenom iba samotnými políčkami lesa, ale na stavoch (*políčko, úroveň zimy*).

Aby riešenie bežalo dostatočne rýchlo, potrebujeme prehľadávanie šikovne orezať. Užitočné pozorovanie je, že ak z fronty používanej pri BFS vytiahneme políčko s horšou úrovňou zimy a horším časom, než sme už na tomto políčku videli, nemá zmysel z neho pokračovať ďalej a môžeme ho zahodiť. To isté políčko má zmysel prehľadávať v neskoršom čase už iba s nižšou doterajšou hodnotou zimy (t. j. vyššou zostávajúcou toleranciou) ako v nejakom skoršom čase, inak je práve vytiahnutý stav ostro horší ako ten minulý.

Keďže stavy vyberáme z fronty podľa času vzostupne, stačí nám pre každé políčko vedieť, s akou najnižšou hodnotou zimy sme na tomto políčku už boli. Ak neskôr vytiahneme z fronty to isté políčko, bude to iste v rovnakom alebo neskoršom čase, takže nás zaujíma jedine, ak máme nižšiu hodnotu zimy. Len v takom prípade políčko spracujeme a pridáme jeho susedov do fronty.

Toto už stačí, aby náš program dobehol v rozumne krátkom čase.

Ďalšie zlepšenie vieme dosiahnuť tak, že si Dijkstrovým algoritmom predpočítame pre každé políčko, aký je najnižší súčet úrovní zimy na nejakej ceste (ľubovoľne dlhej, nie nutne najkratšej) z neho do cieľa (implementácia prehľadávania beží opačným smerom od cieľa, pretože Dijkstrov algoritmus hľadá cesty z jedného vrcholu do všetkých). Ako dĺžka hrany z políčka A na políčko B je teda použitá úroveň zimy na B .

Ak potom naše BFS vytiahne políčko p s doterajšou hodnotou zimy x a cesta s najmenšou celkovou úrovňou zimy z p do cieľa je väčšia ako $Z - x$ (to, čo nám ostáva z tolerancie), tak z tohto stavu nemá zmysel pokračovať v prehľadávaní, žiadnu cestu s vhodnou úrovňou zimy nenájdeme. Budeme v podstate dúfať, že toto políčko ešte niekedy vytiahneme z fronty s horším časom, ale s nižšou úrovňou zimy (alebo pôjdeme inakade).

Výsledné heslo: 513

Vzorové riešenie v C++ nájdete tu: [📄 nocnyLes-solution.cpp](#).