

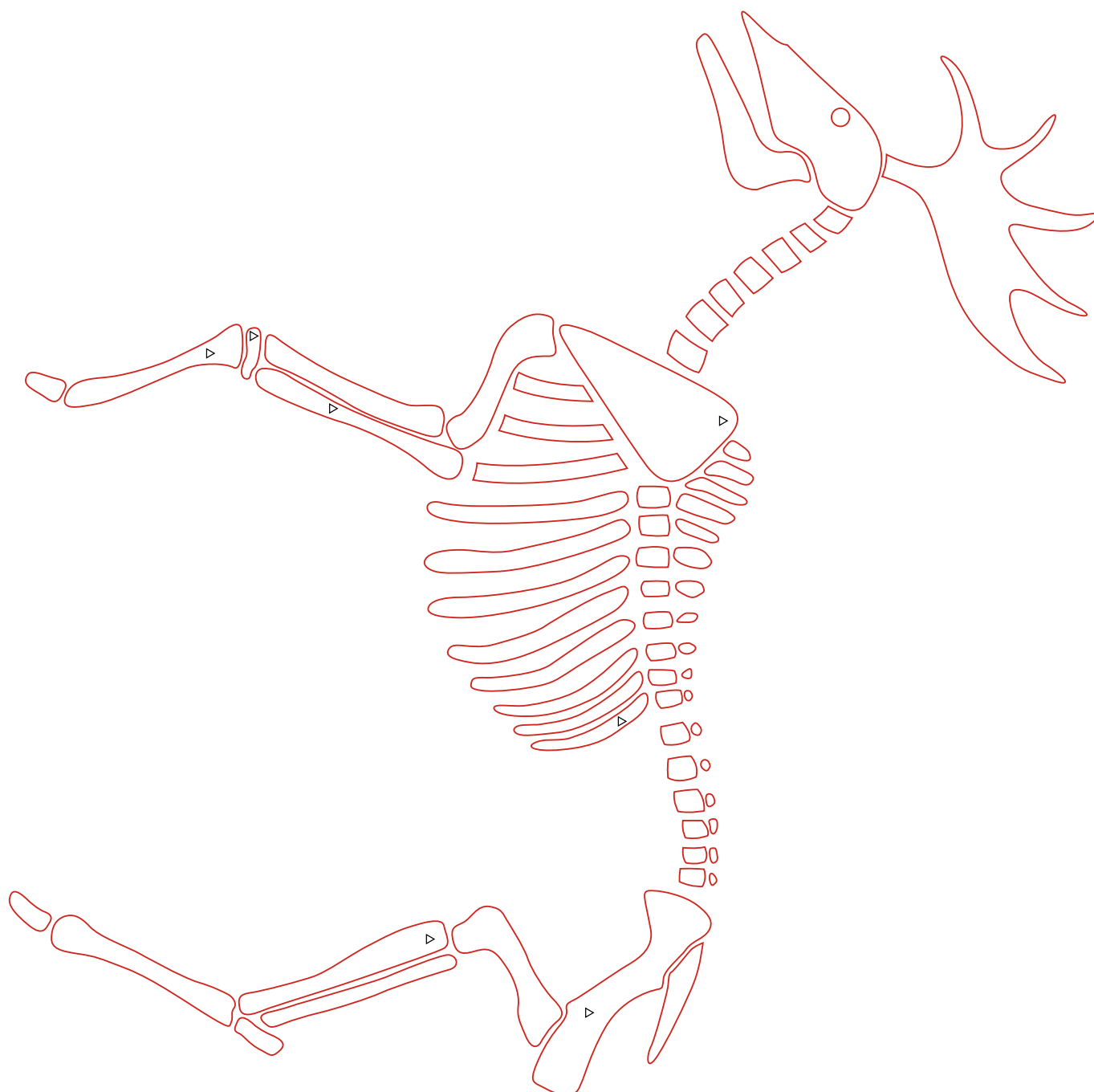
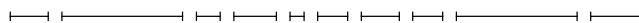


# S1 Lostra



InterLoS 2018

*(Šifra má dvě strany a na druhou se nevešla hlavička, jen číslo stránky, v tom nehledejte součást šifry.)*



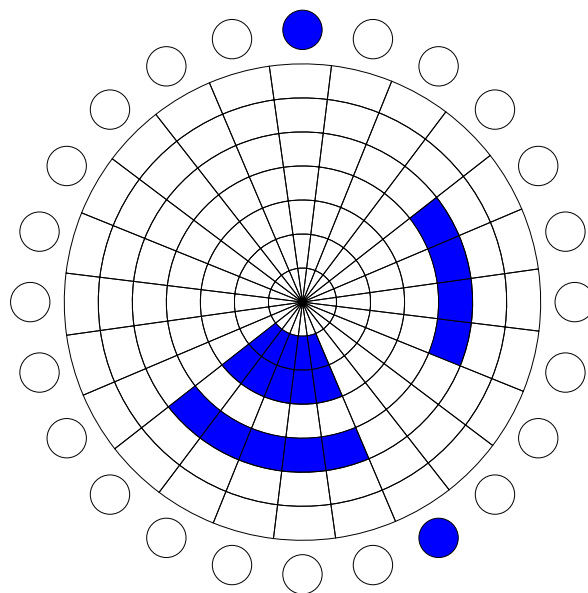
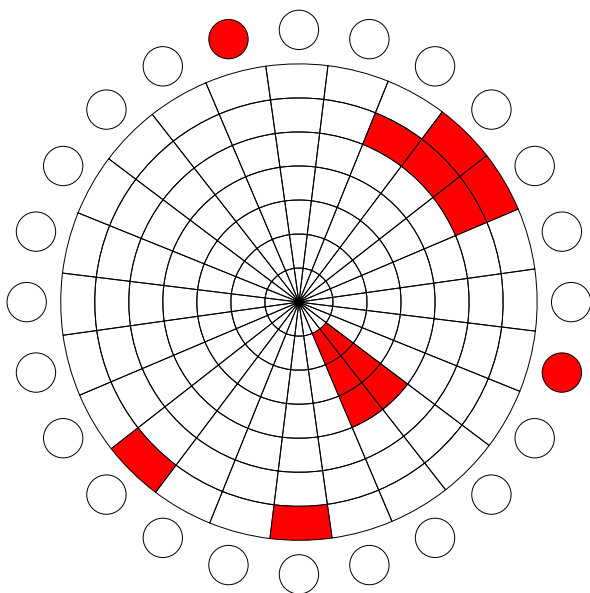
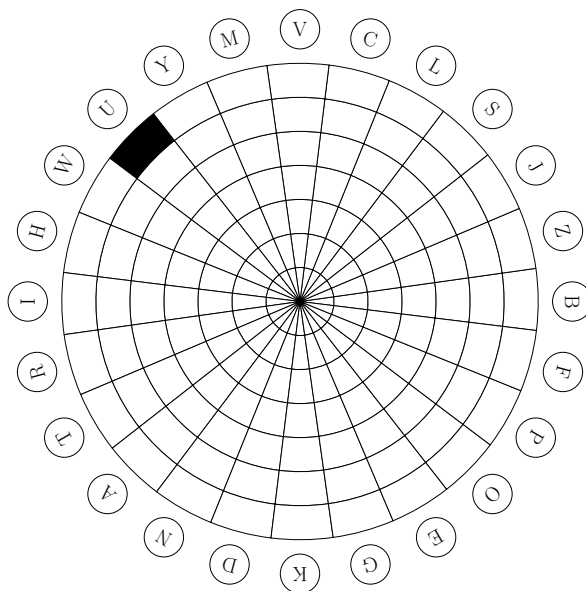




# S2 Pestrofarebná



InterLoS 2018

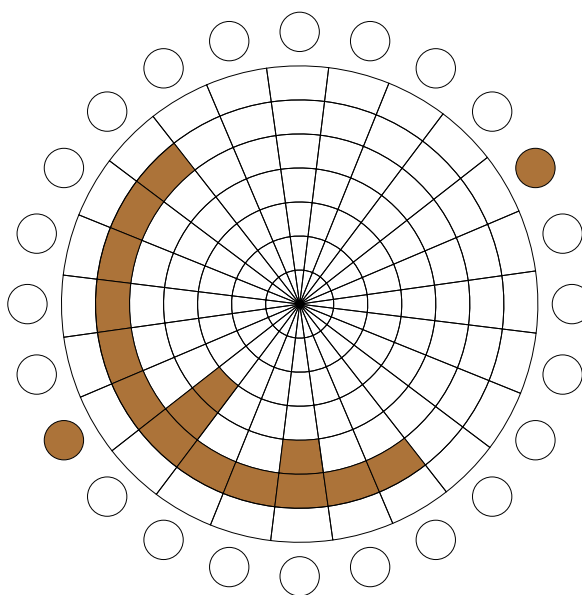
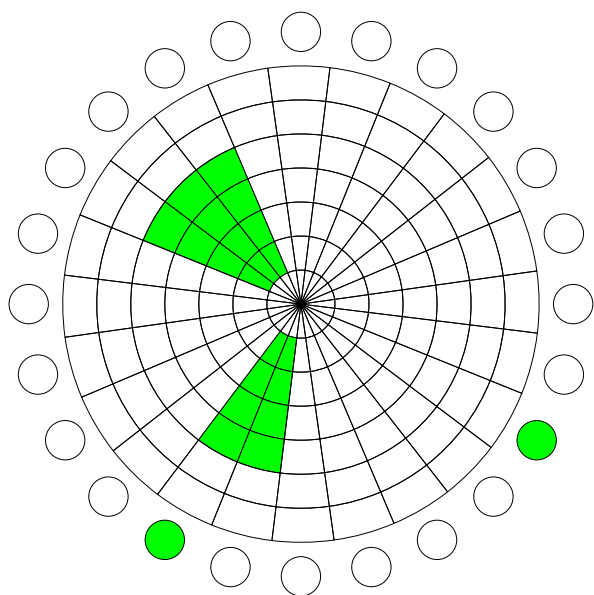
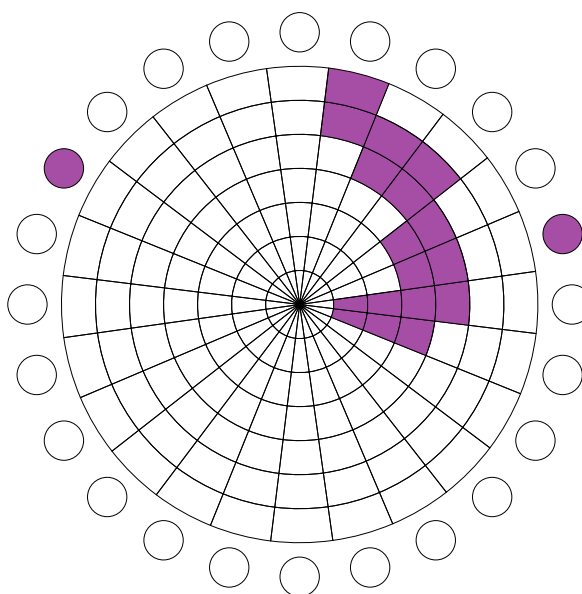
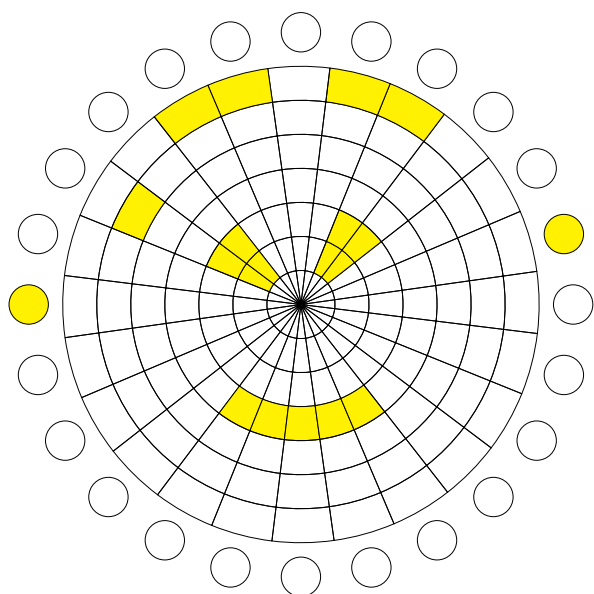




## S2 Pestrofarebná (pokračování)



InterLoS 2018

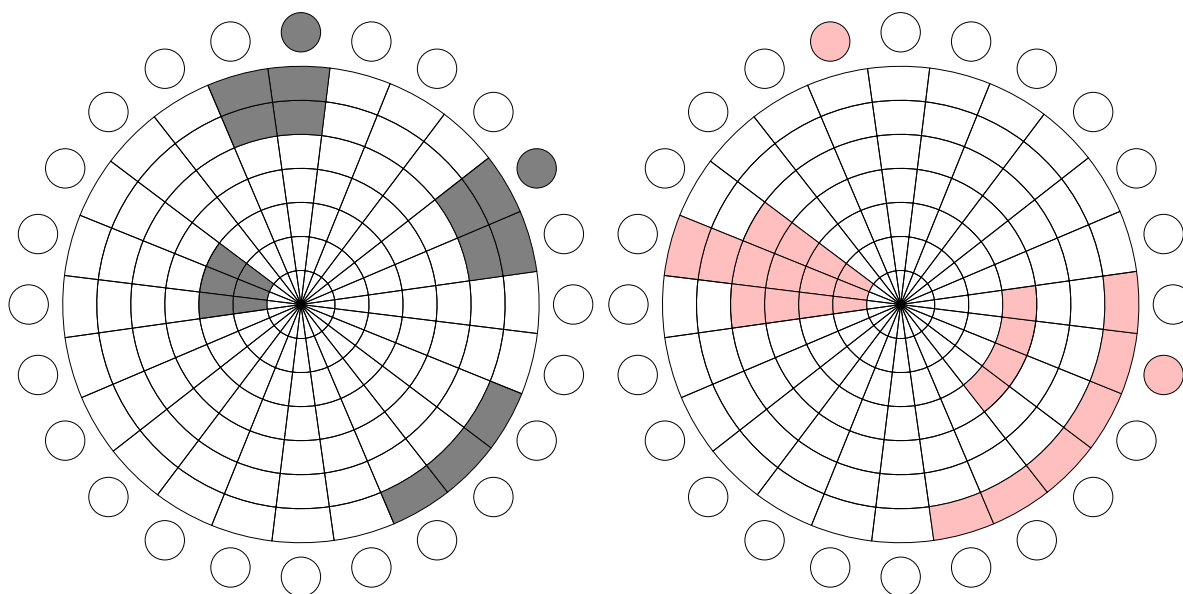




## S2 Pestrofarebná (pokračování)



InterLoS 2018



## S3 Systematická



InterLoS 2018

Úlohu najdete na následující adrese:

<https://interlos.yavanna.cz/systematicka/>

*Poznámka: samotný text hlášek není ani tak šifra, jako spíš instrukce.*



# L1 Los Fillets



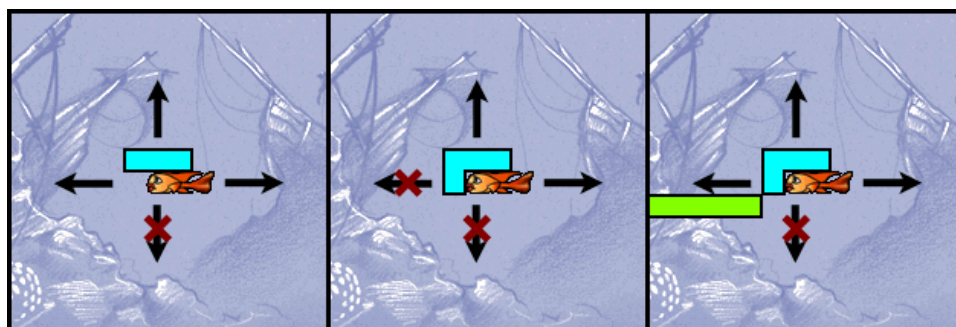
InterLoS 2018

Cílem úlohy je dostat obě rybičky ven. Abyste toho dosáhli, musíte různě přerovnat a přeskupit předměty v místnosti. Při tom je třeba dbát velké opatrnosti, neboť rybičky jsou stvořeny křehké a je snadné je zahubit.

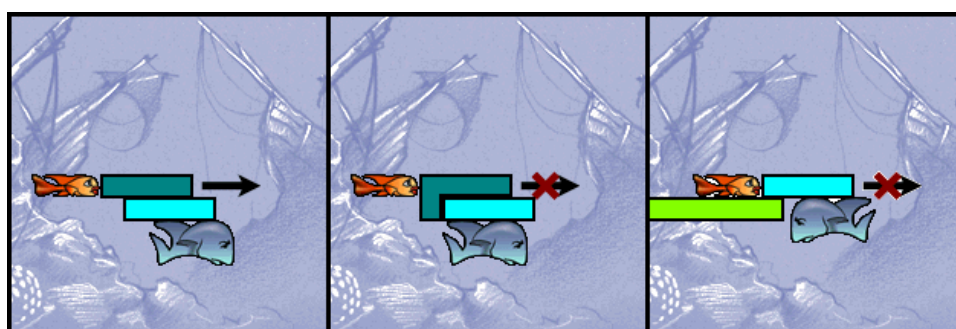
<http://ienze.me/los-fillets/>

Jestliže rybička před sebou nějaký předmět tlačí, musí ho mít položený na nějaké podložce.

Rybička může předmět, který se o nic neopírá, posunout jen tehdy, když se o něco opře v nové pozici, jako třeba na obrázku vpravo.



Není možné posunovat předmět po jiné rybičce, stejně jako rybička sama nemůže tlačit předmět, který se o nic neopírá.



Pokud předmět dopadne na rybičku nebo na něco, co se o rybičku opírá, znamená to pro ni smrt.

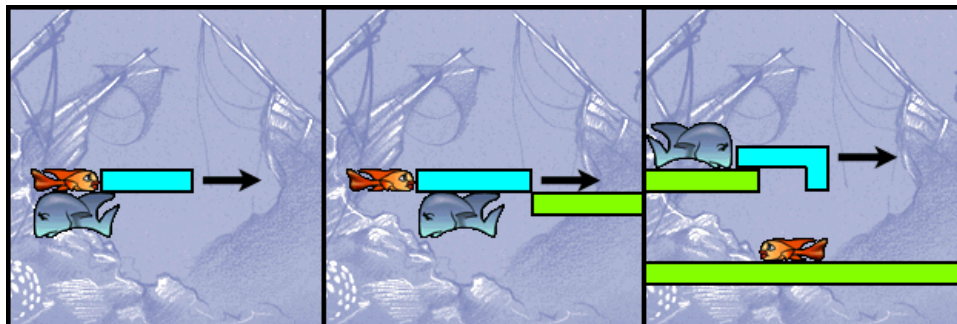
Jediné případy, kdy je možné předmět po rybičce posunout, jsou situace, kdy předmět buď hned spadne nebo se opře o podklad.



## L1 Los Fillets (pokračování)



InterLoS 2018



Rybičky můžete ovládat kurzorovými klávesami a mezerníkem přepínat mezi rybkami. Nebo můžete přímo ovládat malou rybičku klávesami A, S, D a W velkou rybku klávesami J, K, L a I.

Heslo se vám zpřístupní po dokončení levelu.

Možná jste si vzpomněli na původní hru, Fish Fillets NG. Její kompletní návod najdete na adrese <http://fillets.sourceforge.net/manual.cs.php>.



## L2 Krvavá



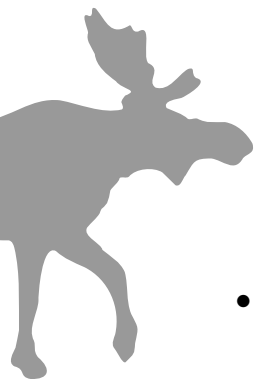
InterLoS 2018

Mladý los Bos je nemocný a je třeba pro něj najít dárce krve. Protože u losů mohou probíhat transfúze jedině v rámci rodiny, budeme hledat mezi jeho příbuznými, kterými jsou Dos, Eos, Ios, Kos, Los, Nos a Sos.

Losí **krvní skupiny** se podobně jako lidské dělí na typy 0 (nula), A, B a AB a podle tzv. Rhesus faktoru (každá ze čtyř skupin má dvě varianty, Rh+ a Rh-, zapisujeme např. jako A+, A-). Uvažujeme tedy osm možných krevních skupin. Každá z krevních skupin značí nějakou kombinaci antigenů A a B a jejich protilátek obsaženou v krvi, což způsobuje, že nemůžeme transfúzi provádět libovolně, reakce antigenu a jeho protilátky totiž vede ke shlukování a rozpadu krvinek.

Možné **darování krve** lze popsat těmito pravidly (pozor, vztahy nejsou vždy symetrické):

- každý z typů 0, A, B, AB může darovat krev stejnému typu, jako je sám, a také ji od něj přijmout (pokud bychom nepočítali s Rh faktorem);
- typy A a B mohou navíc darovat krev skupině AB, typ 0 může dávat kterémukoliv typu;
- nadto je ale třeba dávat pozor na Rh faktor, neboť los s variantou Rh- může dávat negativním i pozitivním variantám, ale Rh+ smí krev darovat pouze opět Rh+;



## L2 Krvavá (pokračování)



InterLoS 2018

- například A<sup>-</sup> tedy může dát krev variantám A<sup>-</sup>, A<sup>+</sup>, AB<sup>-</sup> a AB<sup>+</sup>, ale A<sup>+</sup> jen A<sup>+</sup> a AB<sup>+</sup>;
- žádné jiné možnosti dárce a příjemce nejsou.

Dědění Rh faktoru je jednoduché, potomek může mít kteroukoliv jeho variantu (pozitivní, negativní), která se vyskytuje u jeho rodičů. **Dědění krevní skupiny** ovšem funguje trochu jinak – to, jaká bude výsledná skupina, záleží na kombinaci alel (jsou tři a odpovídají typu A, B a 0) předaných potomkovi oběma rodiči a na jejich dominanci. Můžeme ale přesně popsat, jaké krevní typy potomek určitých rodičů určitě může či nemůže mít:

- pokud mezi rodiči není AB, potomek může vždy mít krevní typ 0;
- pokud je jeden z rodičů A nebo AB, potomek může být A, podobně pokud je mezi rodiči B nebo AB, potomek může být B;
- potomek může získat skupinu AB, pokud je jeden z rodičů AB a druhý není 0, a také když je jeden z rodičů A a druhý B;
- žádné další možnosti zdědění skupiny nejsou.

Každý z našich osmi losů v rodině má jinou z osmi variant krevní skupiny (kombinaci Rh faktoru a jednoho z typů 0, A, B, AB). Jejich rodinné vazby jsou následující: Sos a Kos jsou rodiče Nosa, Los a Ios jsou rodiče Dosa a Eosa, Nos a Dos jsou rodiče Bosa. Žádný los nemůže dát krev sám sobě. Dále po již provedených testech víme, že:

- los Nos může darovat krev právě třem dalším losům, ale ne Eosovi;
- losi Dos a Ios mohou dát krev stejnému počtu příjemců;
- losi Dos a Eos se liší jen Rh faktorem, typ krevní skupiny (0, A, B, AB) je stejný;
- los Ios má faktor Rh<sup>-</sup> a může přijmout krev od Sosa, ale ne od Kosa;
- los Los může přijmout krev od Kosa, ale nesdílí s ním typ krevní skupiny (0, A, B nebo AB).

Došel nám však materiál pro další testování kompatibility. Určete, který z losů může darovat krev Bosovi (v případě více možných dárců vyberte toho nejdřív v abecedě) a také krevní skupiny Bosa i tohoto dárce.

Heslo zapište ve tvaru XYZ, kde:

- X je krevní skupina Bosa: jeden z typů 0, A, B nebo AB následovaný písmenem podle Rh faktoru (P – pozitivní, nebo N – negativní),
- Y je jméno zvoleného dárce velkými písmeny,
- Z je krevní skupina dárce zapsaná stejným způsobem jako Bosova.

Odpověď tedy může vypadat např. takto: *ABNDOS0N*.





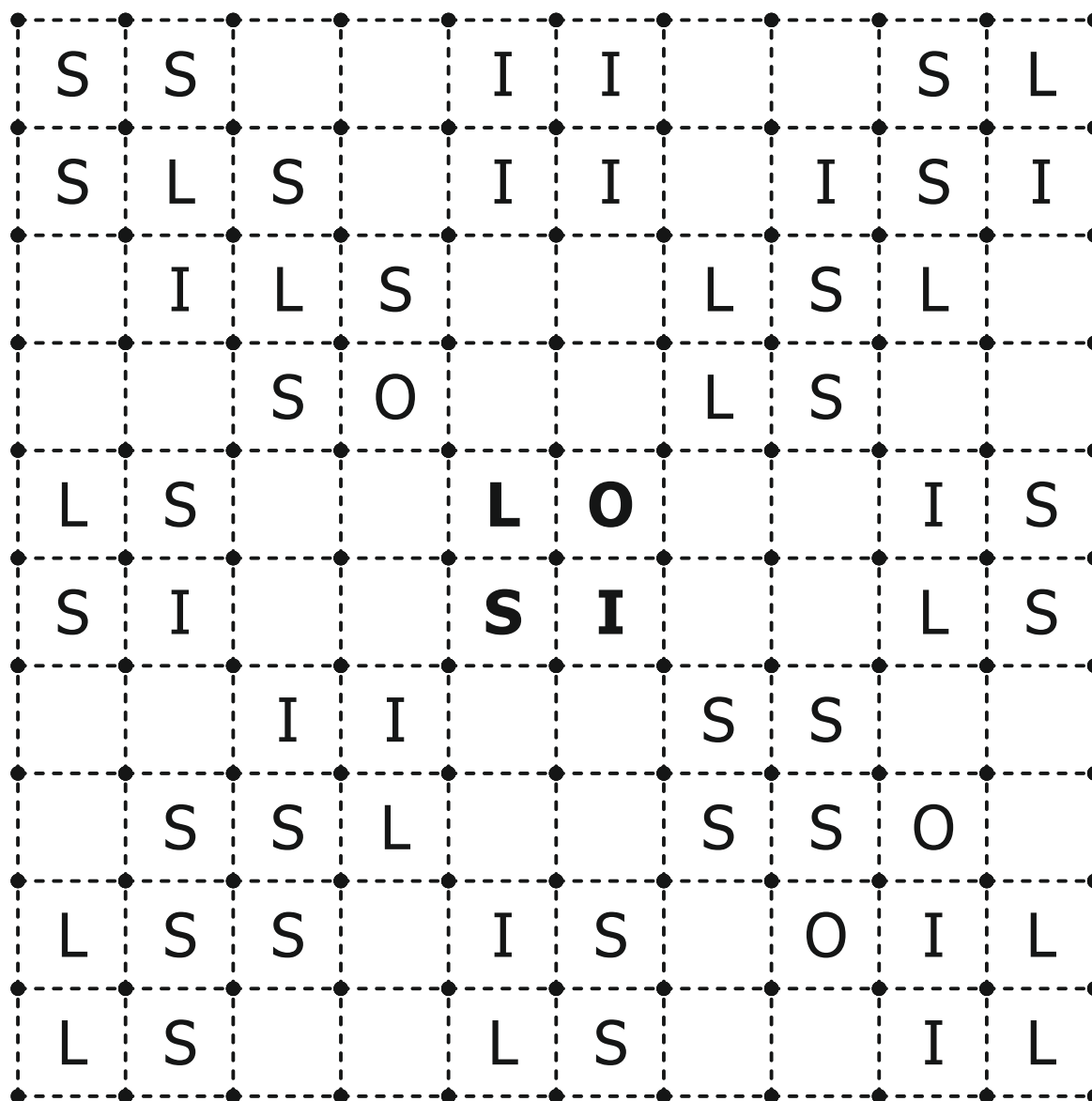
## L3 Ohrádka pro LOSI



InterLoS 2018

Losátka nám od poslední vyrostla. Nahrďte písmena L, O, S, I vhodnými čísly tak, aby stejná písmena v celém obrazci představovala stejné číslo a různá písmena různé číslo. Pak nakreslete jednu uzavřenou smyčku (ohrádku pro losy), aby platilo, že všechna zadaná (zakódovaná) čísla udávají počet stran odpovídajícího čtverce použitých jako plot (každá strana čtverce buď je součástí ohrádky, nebo není). Pokud je čtverec prázdný, není kladeno omezení na to, kolik z jeho stran patří do ohrádky. Pozor, smyčka se nikde nesmí dotýkat sama sebe ani se se sebou křížit.

Jako odpovědní kód zadejte pro každý řádek shora dolů počet „S“ uvnitř ohrádky.





# P1 Jednoduchá



InterLoS 2018

Umíte počítat? Jestli jo, tak už máte tuhle úlohu skoro hotovou. Zbývá jen přesvědčit jednoduchý počítač, že to fakt umíte.

Náš počítač je ale velmi starý a používá feritovou paměť, která má tu super vlastnost, že když nějaký bit paměti přečtete, tak se tím zároveň vynuluje. Navíc se programuje pomocí děrných štítků, tak jsme původně mysleli, že vám nějaké pošleme s pozvánkou a vy nám je pak pošlete zpět s řešením. Jenže pošta by je asi nestihla doručit do konce hry, tak jsme vám to trochu zjednodušili.

Počítač má paměť tvořenou maticí bitů s nějakým počtem řádků a sloupců. Pokud chceme z nějakého místa paměti číst nebo do něj zapisovat, musíme nejprve nastavit adresu bitu – jeho řádek a sloupec. Následně můžeme zapsat 0 nebo 1. Nebo můžeme číst a podle toho, jaká hodnota byla přečtena, se přesunout v programu. Program je posloupnost instrukcí, které mohou mít žádný, jeden, nebo dva argumenty. Počítač je vždy na nějakém místě v této posloupnosti a spouští instrukci na tomto místě. Pokud tato instrukce neurčí jinak, posune se pak počítač na další instrukci. Vysvětleme si instrukce a snad to bude jasné.

instrukce	počet arg.	popis
jump	1	Posune program o daný počet instrukcí (může být záporný). Příklad: <code>jump 0</code> je nekonečný cyklus (neustále zůstává na místě), <code>jump -3</code> skočí o 3 instrukce zpět.
read	2	Přečte hodnotu na nastavené adrese paměťové jednotky a pokud je tato hodnota 0, skočí o hodnotu prvního argumentu, jinak o hodnotu druhého. Pozor, zároveň vynuluje hodnotu v paměti. Příklad: <code>read 3 -4</code> skočí o 3 instrukce dopředu, pokud je v paměti na aktuální adrese 1, jinak o 4 zpět.
write0	0	Zapíše na aktuální adresu 0.
write1	0	Zapíše na aktuální adresu 1.
srow	1	( <i>set row</i> ) Nastaví řádek v adrese na svůj argument (tedy další čtení/zápis půjde z tohoto řádku). Musí být v mezích paměti, jinak počítač shoří.
scol	1	Nastaví sloupec v adrese na svůj argument. Musí být v mezích paměti, jinak počítač shoří.
arow	1	( <i>advance row</i> ) Posune řádek v adrese o argument (může být záporný). Pokud by se měl posunout mimo paměť, objeví se znovu na začátku (přečte modulo počet řádků).
acol	1	Jako <code>arow</code> , ale pro sloupec.
trow	2	( <i>test row</i> ) Otestuje, zda je řádková adresa rovna prvnímu argumentu, a pokud ano, tak skočí o hodnotu druhého argumentu (pokud není, pokračuje další instrukcí).
tcol	2	Obdobně pro sloupec.
end	0	Ukončí program.



## P1 Jednoduchá (pokračování)



InterLoS 2018

Jak jednoduché, člověk z toho úplně vidí, jak se bude ten děrný štítek s instrukcemi posouvat. A přitom stačí, aby ten počítač měl dost velkou paměť a spočítá vám všechno, co jakýkoli jiný.<sup>1</sup>

Tak, a můžeme jít sčítat. Budeme pracovat s pamětí o třech řádcích a libovolném počtu sloupců. Počet sloupců odpovídá počtu bitů čísel. Nejnižší bit čísla je ten se sloupcovou adresou 0. První dva řádky jsou sčítance, do posledního chceme dostat součet. Sčítání binárních čísel funguje úplně stejně jak takové to písemné sčítání, které jste se učili v první (nebo možná druhé) třídě, jen s tím rozdílem, že máte jen číslice 0 a 1 a že pokud by nám zbylo něco po sečtení posledních dvou číslic/bitů, tak to ignorujeme (přetok). Tedy například čtyřbitová čísla 0011 (dekadicky 3) a 0001 (1) dají výsledek 0100 (4). Dále třeba (opět na 4 bitech)  $1001 + 1001 = 0010$ ,  $1111 + 1 = 0000$ ,  $0100 + 0101 = 1001$ .

Napište tedy do <https://hesperia.fi.muni.cz/los/2018/jednoducha/> program, který implementuje sčítání pro libovolný počet sloupců paměti. Login je *interlos* a heslo je *2018*. Pokud váš program bude fungovat, dostanete heslo. Pokud ne, ukáže se vám, jak se program spouštěl, paměť se vypisuje jako matice nul a jedniček se sloupcovou adresou 0 vpravo a řádkovou adresou 0 nahoře. Vidíte vždy instrukci (a její index řádku ve zdrojovém kódu), následně výpis paměti po jejím spuštění a aktuální adresu v paměti (také po spuštění instrukce).



## P2 Uhádni cestu



InterLoS 2018

Migrace losů po světě může většině lidí připadat nepochopitelná, ale vědci z FI MU odhalili, že odpovídá matematickým zákonitostem, a to zejména v tom, že místa prochází pouze v jednom směru bez opakování. Pojďme prověřit toto zjištění na základě záznamů o pohybu losů.

Vstupem této úlohy bude pole tras, kde každá trasa odpovídá posloupnosti čísel vrcholů. Vaším cílem je co nejpřesněji odhalit graf, po kterém se losi pohybují. K tomu ale nebudete mít dostatek dat, takže se budete muset snažit o co nejpřesnější dolní a horní odhad.

Dne 8. 12. 2018 fotopasti zachytily nějaké losy na sledovaných vrcholech v pořadí, které dostanete jako vstup `order`. Nicméně nejsme schopni rozlišit, která fotopast zachytila jakého losa, pouze víme, že losí král Jørgen v tento den začal ve vrcholu 0 a skončil ve vrcholu  $x$ . Řešením úlohy je odpověď na otázku „Kolika nejvíce a nejméně vrcholy mohl Jørgen projít?“. Heslo je jedno číslo ve tvaru `<min><max>` zřetězeným za sebou bez mezery.

<sup>1</sup> Zvídaví si mohou za 0 bonusových bodů dokázat, že pro rozměry paměti alespoň 1 krát nekonečno dokážeme pro libovolný Turingův stroj najít ekvivalentní program pro náš počítač (který akceptuje instrukci `end` a zamítá shořením).



## P2 Uhádni cestu (pokračování)



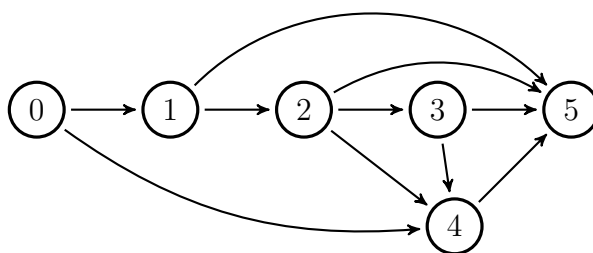
InterLoS 2018

**Poznámka** Pořadí `order` slouží k tomu, aby ještě více omezilo odhady grafu. A to tak, abyste nehledali nejdelší cestu v cyklickém grafu.

**Vstup** Soubor `P2-input.json` obsahuje slovník ve formátu JSON. Ten má v kořeni 4 atributy:

- `nodes_count` říká, kolik vrcholů má graf, se kterým pracujeme. Vrcholy grafu číslujeme od 0 do `nodes_count - 1`,
- `traces` je pole tras = posloupností vrcholů,
- `order` je uspořádání, ve kterém fotopasti zachytily losy a
- `target` je cíl  $x$ , do kterého hledáte trasu z vrcholu 0.

**Příklad** Skutečný graf, který nám není znám, může vypadat nějak takto:



Váš vstup může být následující:

- `nodes_count`: 6
- `traces`: (2, 3, 5), (1, 2, 4), (1, 2, 3, 4, 5) a (2, 5)
- `order`: (0, 4, 1, 3, 2, 5)
- `target`: 5

Kdybychom znali graf uvedený výše, byla by nejdelší a nejkratší cesta vzhledem k pořadí `order` zřejmá:

- nejdelší: 0, 4, 1, 3, 2, 5 – délky 4 a
- nejkratší: 0, 4, 1, 3, 2, 5 – délky 3.

Vy ale musíte prvně najít grafy dolního a horního odhadu.

Při hledání nejdelší možné cesty budujeme horní odhad grafu. Ten bude obsahovat všechny hrany, jejichž existence není vyloučena pořadím v zachycených trasách. Například z trasy (2, 3, 5) vidíte, že v grafu nemůže existovat hrana  $5 \rightarrow 3$  ani tranzitivně  $5 \rightarrow 2$ . Ale stále je možné tuto trasu proložit dalšími vrcholy (jak je vidět v trase (1, 2, 3, 4, 5)) nebo nějaké vrcholy odebrat, viz trasa (2, 5) – proto jsme mohli odstranit pouze 2 hrany.

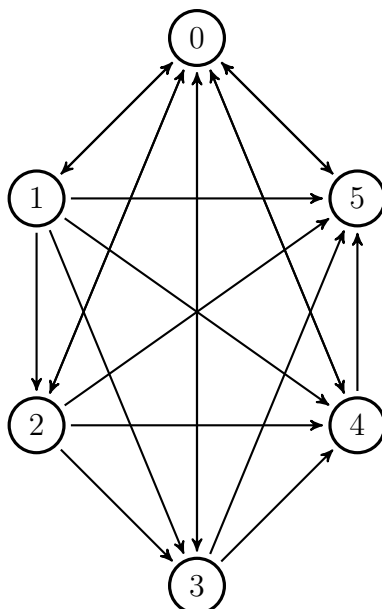


## P2 Uhádni cestu (pokračování)



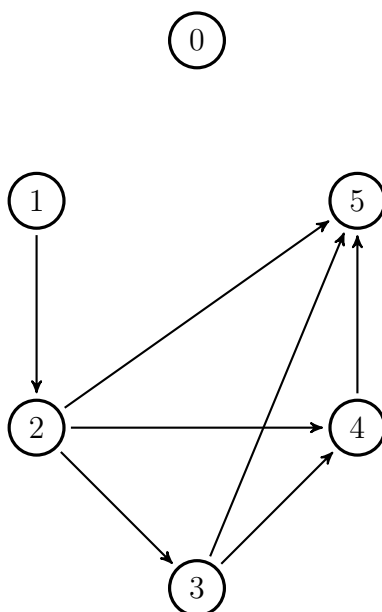
InterLoS 2018

Graf horního odhadu pro zadaný vstup vypadá takto:



S ohledem na zadané pořadí **order** jsou 2 nejdelší cesty z vrcholu 0 do vrcholu 5 v grafu:  $0 \rightarrow 1 \rightarrow 3 \rightarrow 5$  a  $0 \rightarrow 1 \rightarrow 2 \rightarrow 5$ , tedy délky 3.

Druhý úkol je hledání nejkratší cesty. Potřebujeme tedy zkonstruovat dolní odhad grafu tvořený hranami z tras. Například z trasy  $(2, 3, 5)$  víme, že graf bude obsahovat hrany  $2 \rightarrow 3$  a  $3 \rightarrow 5$ . Existenci tranzitivních hran nám trasy nedávají! Vybudovaný graf pro zadané trasy vypadá následovně:





## P2 Uhádni cestu (pokračování)



InterLoS 2018

Cesta z vrcholu 0 do vrcholu 5 by v tomto případě neexistovala. Kdybychom měli hledat cestu z vrcholu 1, tak by s ohledem na zadané pořadí *order* šlo o cestu  $1 \rightarrow 2 \rightarrow 5$ . Všimněte si, že cesta  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$  by porušila pořadí *order*, problém je s hranou  $2 \rightarrow 4$ .



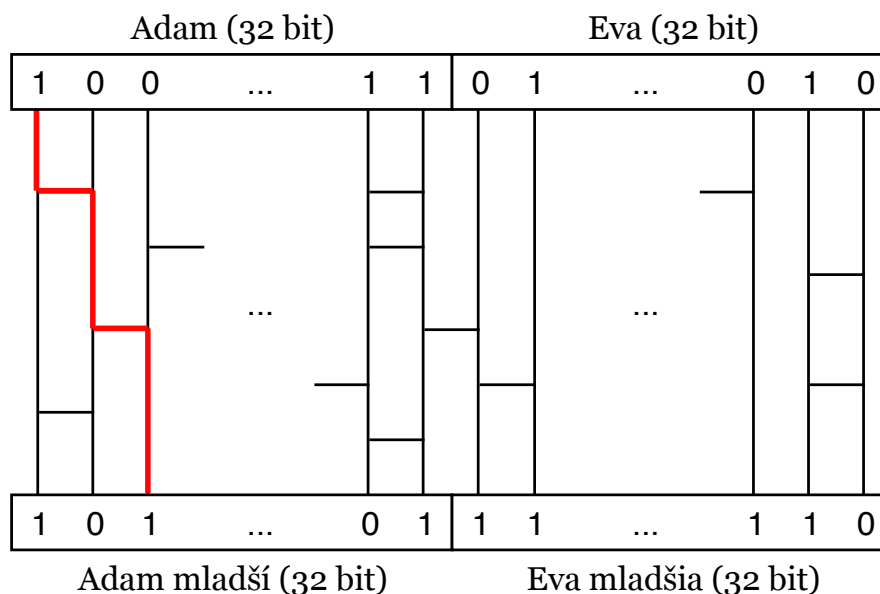
## P3 Nezvestná



InterLoS 2018

Každý *Adam* a *Eva* sa dá reprezentovať ako celé číslo (postupnosť 32 bitov).

*Adam* a *Eva* sa vždy krížia tým istým spôsobom, ktorý je popísaný tabuľkou `P3-input.txt`. 32 bitov popisujúcich *Adama* a 32 bitov popisujúcich *Evu* sa zaradia za seba. Potom každý bit prebehne tabuľkou kríženia. Začína v odpovedajúcom stĺpci a nasleduje zvislú čiaru. Ak narazí na nejakú horizontálnu priečku spájajúcu zvislé čiary, preskočí na vedľajšiu (tú druhú, s ktorou je spojená horizontálna priečka). Keď takto všetky bity prebehnú, rozdelia sa na dve časti po 32 bitov, ktoré určujú novú generáciu *Adama* a *Evy*.



Tento rok sa stalo niečo úžasné. Vznikli dokonalé dvojčky  $Adam_{42}$  a  $Eva_{42}$ .

Vieme, že pôvodný *Adam* bol  $Adam_{16641}$ . Nájdite takú *Evu* pre pôvodného *Adama*, aby po mnohých generáciách vznikli dokonalé dvojčky. Číslo *Evy* je výsledné heslo.

Ak by ste si chceli overiť, že čítate tabuľku kríženia správne, vieme, že potomkami  $Adam_{42}$  a  $Eva_{42}$  budú  $Adam_{136}$  a  $Eva_{67174720}$ .