

Domácí úkol 2 - Grafová hra

Termín odevzdání: Nejpozději na cvičení 9. května

Vaším úkolem je zanalyzovat následující kombinatorickou hru a navrhnout vyhrávající strategii.

Herní plán má podobu orientovaného grafu (viz obrázek na konci), hrají dva hráči - *červený* a *zelený*. Jedna partie odpovídá nekonečnému (orientovanému) sledu v tomto grafu. Ve vrcholu označeném červenou barvou je na tahu *červený hráč*. Může zvolit jeden z tahů A, B, C, D . V zeleném vrcholu je na tahu *zelený hráč* a může zvolit jeden z tahů $1, 2, 3, 4$. Hráči se v tazích střídají a postupně vytváří nekonečný sled v grafu. Začíná *červený hráč* v červeném vrcholu.

Zaveďme následující značení:

- $G = (V, E)$ - graf, na kterém se hraje
- w - nekonečný (orientovaný) sled v grafu G . Odpovídá jedné partii.
- C - množina tahů červeného hráče. $C = \{A, B, C, D\} \subset V$
- Z - množina tahů zeleného hráče. $Z = \{1, 2, 3, 4\} \subset V$
- $visits_w$ - funkce, která každému vrcholu v v grafu G přiřazuje číslo podle toho, kolikrát sled w navštívil tento vrchol.
 $visits_w : V \mapsto (\mathbb{N} \cup \{0\} \cup \infty)$
 $visits_w(v)$ = počet výskytů vrcholu v ve sledu w .
- $C^\infty(w) = \{x \in C \mid visits_w(x) = \infty\}$ - množina vrcholů z C , které se ve w vyskytují nekonečně-krát.
- $Z^\infty(w) = \{x \in Z \mid visits_w(x) = \infty\}$ - množina vrcholů ze Z , které se ve w vyskytují nekonečně-krát.
- $N(w) = \max(Z^\infty(w))$ - největší z čísel, které se ve w vyskytují nekonečně-krát.

Bud' w jedna partie hry. řekneme, že *zelený hráč vyhrál* partii w , pokud platí $|C^\infty(w)| = N(w)$ - čili největší číslo ze Z , který se vyskytuje nekonečně často je rovné počtu vrcholů z C , které se ve w vyskytují nekonečně často. V opačném případě řekneme, že *červený hráč vyhrál* partii w .

Příklad 1:

- *červený hráč* hraje vždy tah A , *zelený hráč* hraje vždy tah 1
- hra vypadá takto: $w = A, 1, A, 1, A, 1, A, 1, \dots$ (vrcholy, které nejsou v C ani v Z není potřeba zmiňovat)
- $C^\infty(w) = \{A\}$, $|C^\infty(w)| = 1$
- $Z^\infty(w) = \{1\}$, $N(w) = 1$
- *zelený hráč* vyhrál, protože $N(w) = |C^\infty(w)|$.

Příklad 2:

- červený hráč hraje vždy tah B , zelený hráč hraje vždy tah 2
- hra vypadá takto: $w = B, 2, B, 2, B, 2, \dots$
- $C^\infty(w) = \{B\}$, $|C^\infty(w)| = 1$
- $Z^\infty(w) = \{2\}$, $N(w) = 2$
- červený hráč vyhrál, protože $N(w) \neq |C^\infty(w)|$.

Příklad 3:

- červený hráč zahraje dvakrát D a pak střídá tahy A, C
- zelený hráč zahraje třikrát 4 a pak střídá tahy $3, 1$
- hra vypadá takto: $w = D, 4, D, 4, A, 4, C, 3, A, 1, C, 3, A, 1, C, 3, A, 1, C, 3, \dots$
- $C^\infty(w) = \{A, C\}$, $|C^\infty(w)| = 2$
- $Z^\infty(w) = \{1, 3\}$, $N(w) = 3$
- červený hráč vyhrál, protože $N(w) \neq |C^\infty(w)|$.

Příklad 4:

- červený hráč hraje postupně A, B, C, D , zelený hráč hraje vždy tah 4
- hra vypadá takto: $w = A, 4, B, 4, C, 4, D, 4, A, 4, B, 4, C, 4, D, 4, \dots$
- $C^\infty(w) = \{A, B, C, D\}$, $|C^\infty(w)| = 4$
- $Z^\infty(w) = \{4\}$, $N(w) = 4$
- zelený hráč vyhrál, protože $N(w) = |C^\infty(w)|$.

Strategie hráče je *program*, který na vstupu postupně dostává tahy protivníka a na výstup postupně vypisuje tahy hráče. *Strategií* si můžeme obecně představit takto:

```
1 program Strategie;
2 var ... zde můžou být proměnné a datové struktury
3 begin
4   while true do
5     read(tahProtivnika);
6     mujTah := zvolMujTah(tahProtivnika);
7     write(mujTah);
8 end.
```

Jak konkrétně bude strategie hrát závisí na implementaci funkce *zvolMujTah*.

Příklad 5:

- červený hráč používá strategii $S1$: Pokud protivník zahrál dvakrát po sobě stejně, zahraj **B**, jinak zahraj **A**
- zelený hráč používá strategii $S2$: Pokud protivník zahrál dvakrát po sobě stejně, zahraj **1**, jinak zahraj **2**
- hra vypadá takto: $w = A, 2, A, 1, A, 1, B, 2, A, 2, B, 2, B, 1, A, 2, A, 1, \dots$
- $C^\infty(w) = \{A, B\}$, $|C^\infty(w)| = 2$
- $Z^\infty(w) = \{1, 2\}$, $N(w) = 2$
- zelený hráč vyhrál, protože $N(w) = |C^\infty(w)|$. říkáme, že strategie $S2$ poráží strategii $S1$.

Příklad 6:

- červený hráč používá strategii $S3$: $X =$ číslo, které protivník hrál nejčastěji. Pokud $X \leq 2$, zahraj náhodný tah z množiny $\{A, B, C, D\}$, jinak zahraj náhodný tah z množiny $\{A, B\}$.
- zelený hráč používá strategii $S2$ stejnou jako minule.
- hra může vypadat např. takto: $w = A, 2, C, 2, C, 1, B, 2, D, 2, B, 2, A, 2, A, 1, D, 2, \dots$
- $C^\infty(w) = \{A, B, C, D\}$, $|C^\infty(w)| = 4$
- $Z^\infty(w) = \{1, 2\}$, $N(w) = 2$
- červený hráč vyhrál, protože $N(w) \neq |C^\infty(w)|$. Strategie $S3$ poráží strategii $S2$.

Strategie se nazývá *vyhrávající* pokud poráží jakoukoliv strategii protivníka.

Vaše úkoly a bodové ohodnocení:

- Popište složitost strategií $S1$, $S2$, $S3$. (Viz níže.) [10 bodů]
- Najděte strategii, která poráží strategii $S3$ z příkladu 6 a popište její složitost. [5 bodů]
- Rozhodněte, jestli má některý z hráčů vyhrávající strategii v této hře. Pokud ne, dokažte, že neexistuje, pokud ano, najděte ji, dokažte, že je vyhrávající a napište příslušný program (stačí v pseudokódu). [20 bodů]
- Popište časovou a paměťovou složitost funkce *zvolMujTah* (ve vámi navržené vyhrávající strategii). Popište, kolik času bude potřebovat jedno zavolání funkce *zvolMujTah* a kolik paměti bude celý program potřebovat v závislosti na čísle M , kde M značí, kolikrát už byla funkce *zvolMujTah* zavolaná. (Např. můžete zjistit, že váš program potřebuje např. $\mathcal{O}(M^2)$ času na M -té zavolání funkce a po M zavoláních funkce program využívá např. $\mathcal{O}(M^3)$ paměti). [15 bodů]

Bonusové úkoly: [celkem 50 bonusových bodů]

- Vyřešte zobecněnou hru, kde velikost množin C a Z nebude předem známá. Množiny budou mít v tomto případě n prvků. Množina C bude obsahovat n (nějakých) symbolů a množina Z bude obsahovat n čísel (od 1 do n). Pravidla hry i definice výhry a prohry zůstávají stejné. (Původní zadání je pro $n=4$). Najděte a popište vyhrávající strategii pro některého z hráčů.

- Popište časovou a paměťovou složitost funkce *zvolMujTah* ve vámi navržené vyhrávající strategii pro zobecněnou hru. Složitost může nyní navíc záviset na čísle n .
- Najděte vyhrávající strategii, jejíž složitost (časová ani paměťová) nezávisí na čísle M . (čili čas na jedno zavolání funkce *zvolMujTah* nezávisí na tom, kolikrát už byla funkce zavolaná, a program po celou dobu běhu vyžaduje paměť závisující pouze na čísle n).

V případě nejasností v zadání mi pošlete e-mail.
řešení můžete posílat buď e-mailem nebo odevzdat na papíře na cvičení.

