

Zápočtový program: Hra Logic (Mastermind)

Martin Průša

Zimní semestr 2024/2025

Programování 1

Zadání

Známa hra logik. Hra je parametrizována dvěma čísly: počet pozic N (typicky 4 nebo 5) a počet barev M (typicky 6 nebo 8). První hráč na počátku zvolí nějakou N -prvkovou kombinaci daných barev (mohou se i opakovat), druhý se ji snaží uhodnout. V každém tahu druhý hráč navrhne nějakou kombinaci barev a první mu na ni odpoví dvěma čísly: počet správně uhodnutých pozic a počet barev, které sice v utajované kombinaci jsou, ale na jiném místě. V programu jde hrát jak za hádajícího hráče, tak za hráče vybírajícího. V případě že hráč hraje za hráče vybírajícího tajnou kombinaci, počítač by ji měl pro typický počet pozic a barev uhodnout za 10 tahů a do jedné minuty.

Zvolený algoritmus

Hádání tajné kombinace počítačem

Popis algoritmu

- Vytvoří se pole A délky N , které bude obsahovat samé 1.
 - Vytvoří se proměnná *change_index*, která reprezentuje počet uhodnutých barev (ne nutně správných pozicích, barvy mohou být i stejné)
1. Pokud ještě není žádná historie hry, program bude chtít zhodnotit pole A .
 2. Jedniček nechá v poli A (na začátku pole) tolik, kolik byl počet správných barev na správných pozicích. Zbytek čísel v poli A zvýším o 1. Program bude chtít zhodnotit náhodné uspořádání pole A . Také zvýší *change_index* o počet uhodnutých barev.
 - Poznámka: Pole A se náhodně uspořádá, protože pak se lépe určí správné pozice jednotlivých barev
 3. Na začátku pole A ponechá beze změny tolik pozic, kolik je uloženo v *change_index*. Zbytek čísel v poli A zvýší o 1. Opět bude chtít zhodnotit náhodné uspořádání pole A a opět zvýší *change_index* o počet uhodnutých barev. Toto se opakuje, dokud *change index* není roven počtu hádaných pozic.
 4. Když už vím, jaké přesně barvy tajná kombinace obsahuje postupně začnu procházet permutace pole A , u každé budu kontrolovat, zda by odpovídala historii. Pokud historii odpovídá, vrátím ji. Toto se opakuje, dokud program neuhodne tajnou kombinaci.

- Poznámka: Odpovídá historii znamená, že počet barev, které jsou na stejných místech jako v aktuální permutaci je roven počtu barev na správné pozici v historickém tahu

Příklad algoritmu (pouze hledání barev)

Počet míst: 5

Počet barev: 8

Tajný kód: 3 5 1 3 2

Jak bude vypadat pole A v jednotlivých krocích:

Hodnocení = počet barev na správných pozicích/ počet správných barev na špatných pozicích

Součet obou složek hodnocení bude v programu stejný, ale rozdělení je náhodné

- 1 1 1 1 1 -> Hodnocení: 1/0 -> Ponechám jednu jedničku
- 1 2 2 2 2 -> Hodnocení: 2/0 -> Ponechám jednu jedničku a jednu dvojku
- 1 2 3 3 3 -> Hodnocení: 1/3 -> Ponechám jedničku, dvojku a dvě trojky
- 1 2 3 3 4 -> Hodnocení: 1/3 -> Ponechám to, co v předchozím kroku
- 1 2 3 3 5 -> Hodnocení: 1/4 -> Mám všechny barvy v kombinaci

Nyní bych se na základě historie snažil najít odpovídající řešení

Diskuse výběru algoritmu

Zvažoval jsem také algoritmus, který by nejprve testoval dvojice barev (vždy polovina barev v tahu by byla jedné barvy, polovina druhé). Když by celkový počet uhodnutých barev byl roven počtu pozic, zkoušel by vytvořit takovou kombinaci, která by odpovídala předchozím hodnocením. Tento algoritmus by byl ovšem složitější na implementaci a počet tahů, na které by tajný kód uhodl by nebyl o tolik menší.

Program

Program lze spustit příkazem `python3 main.py`.

Soubor **main.py** obsahuje hlavní program - co se má kdy vypsát, kdy se má načíst vstup, kdy se má provést další kolo, atd.

Soubor **reader.py** obsahuje třídy představující čtečky vstupů (nyní pouze tu z konzole)

Soubor **printer.py** obsahuje třídy reprezentující tiskárny (nyní pouze tu do konzole)

Soubor **logik.py** obsahuje třídu se samotnou logikou hry logic. Jak probíhá tah, zhodnocení kombinace čísel, atd.

Soubory ve složce **players** představují třídy hráče hry. Nyní hráče ovládaného uživatelem a automatického hráče. Každé třída v této složce by měla mít metodu "get_correct_sorted_combination(self, history)".

Reprezentace vstupních dat a jejich příprava

Jak zadávat vstupy program vždy vypíše, než se zeptá na vstup. Nejzajímavější je:

Kombinace čísel k ohodnocení / tajná kombinace se zadává na jeden řádek, jednotlivá čísla jsou oddělená jednou mezerou. Čísla musí být od 1 do počtu barev ve hře. Čísel musí být tolik, kolik je počet hádaných pozic.

Reprezentace výstupních dat a jejich interpretace

Program vypisuje jednotlivé tahy na nové řádky. Čísla v hádaná v tahu jsou oddělena mezerou. Znakem „|“ je pak odděleno zhodnocení tahu od hádaných čísel.

Hodnocení tahu:

Číslo za P značí počet barev na správných pozicích

Číslo za B značí počet správných barev na špatných pozicích

Ukázka

Číslo tahu									
Hádaná kombinace					Počet barev na správných pozicích				
					Počet správných barev na špatných pozicích				
0:	1	1	1	1	1	P	0	B	0
1:	2	2	2	2	2	P	2	B	0
2:	3	3	2	3	2	P	0	B	2
3:	4	2	2	4	4	P	1	B	3
4:	2	2	4	5	4	P	2	B	3
5:	2	4	5	2	4	P	5	B	0

Počítač uhodl tvou kombinaci
Tajná kombinace byla: 2 4 5 2 4
Počet pokusů: 6
Doba hraní: 0.0008962154388427734 s