

Logik

Zápočtový program na NMIN101, 2012/2013 Joel Jančařík

Anotace

Práce se zabývá hrou Logik neboli Mastermind. Hlavní část programu se zabývá řešením této hry za stranu hádajícího hráče. Cílem bylo nalézt algoritmus, který se v rozumném počtu kroků dopracuje ke správnému řešení, a který není závislý na přesném nastavení barev a počtu políček ve hře. Jako vedlejší produkt vznikla možnost hrát hru Logik za oba hráče.

Přesné zadání

Program bude řešit hru Logik. Počet tahů pro nastavení s osmi barvami a pěti políčky by neměl být vyšší než deset. Program bude obsahovat možnost si hru zahrát za hráče. Program bude čistě řádkový.

Zvolený algoritmus

Použitý algoritmus na hledání řešení se dělí na tři hlavní části. V první fázi se na co nejmenší počet kroků snaží uplatnit všechny barvy. To pro další fáze získává cenné informace o počtu opakujících se barev a lze z toho usoudit maximální počet políček jednotlivých barev. V druhé fázi se hledá jedno řešení, které by mohlo obsahovat stejné barvy jako výsledek. Toto řešení se následně vyzkouší. Do třetí fáze se vstupuje se znalostí barev, které jsou v řešení a jde jen o to je uspořádat do správného pořadí.

V dalším je použito písmeno n pro počet barev a písmeno k pro počet políček

První fáze

Nejprve se spočítá počet tahů nutný k vyzkoušení všech barev jako n děleno k zaokrouhleno nahoru. Poté se spočítá počet barev, které je nutné zdvojit jako $(\text{počet pokusů}) * k$ to celé modulo n . Toto číslo je určitě menší než n a tedy lze příslušný počet barev zdvojit. Poté už jen vytvoří možné pokusy a to tak, aby zdvojené barvy byly rozloženy co nejvíce rovnoměrně.

Druhá fáze

Druhá fáze je asi nejsložitější částí programu. Na vyhledání možného řešení se používají dvě rekurzivní procedury *rekurze* a *vyber*. Rekurze postupuje po již získaných výsledcích a zadává vyber kolik a jakých prvků má z daného výsledku vybrat. Procedura *vyber* poté zkouší postupně všechny kombinace ze zadaných prvků. Pro každou nalezenou kombinaci se zavolá *rekurze* na další výsledek. Ve chvíli, kdy se nalezne možné řešení, tak se vše zastaví a nedokončuje.

Třetí fáze

Ve třetí fázi již známe barvy, které se nalézají v řešení. Dalším úkolem je nalézt možná řešení. Toto se dělá prozkoumáním všech možných uspořádání barev a jejich kontrolou s výsledky předcházejících otázek. Pokud jich je více, pak se z nich vybírá takový, který po svém vyzkoušení v nejhorším případě vyřadí nejvíce ostatních řešení. Toto číslo se dostane tak, že se ostatní rozdělí podle počtu společných barev na jednotlivých políčkách do skupin a zjistí se velikost největší skupiny.

Diskuze výběru algoritmu

Algoritmus byl vybírán tak, aby se co nejvíce přiblížil mému postupu při hraní hry Logik. Problém nastal v druhé fázi, kde jsem zvyklý zkoušet pseudonáhodné vstupy. Zde bylo potřeba zavést určitý řád a ten byl docílen podmínkou, že zadaná otázka by mohla být řešením. Toto jistě není nejlepší řešení, ale zajišťuje konečnost zvoleného algoritmu a navíc je zde předpoklad, že pokud je toto řešení možné, tak se pravému řešení blíží. V druhé fázi není používáno rozdělení na bílé a černé body, zde je určitě prostor pro zlepšení. V třetí fázi se pokouším o co nejrychlejší nalezení výsledku. I zde ovšem vybírám otázku z možných řešení, což opět nemusí být nejrychlejší postup. Navíc je zde nutnost si možná řešení pamatovat a to není vždy umožněno. V případě, že je počet možných řešení větší než konstanta *omezeni*, pak se použitý algoritmus nedá pokládat za dostatečně efektivní.

Program

1. Fáze jedna
2. Druhá fáze
 - a) Rekurze
 - b) Vyber
3. Třetí fáze
 - a) Generuj (generuje všechna řešení o daném složení barev)
 - b) Vyzkoušej (zkouší vygenerovaná řešení, zda jsou možná)
 - c) Spočti (Hledá nejvhodnější řešení)
4. Kontrola

Důležité konstanty: *n*, *k*, *omezeni* (viz. předcházející), *max* (maximální počet tahů)

Důležité proměnné: *pole* (zde je uložený výsledek), *vyzkoušeno* (zde jsou uloženy vyzkoušené vstupy), *výsledky* (zde jsou uloženy výsledky), *d* (počet proběhlých tahů)

Reprezentace vstupních dat a jejich příprava

Program je schopen si vytvářet vlastní vstupní data do souboru (v možnosti C v prvním menu). V případě manuálního zadávání je potřeba dodržet správný počet číslic (*k*-obvykle pět) a jejich velikost od 1 do *n*, kde *n* je obvykle 8. Číslice se píší na řádek a oddělují se mezerou. Na konci řádku odřádkujte. Vstup se nachází ve složce s programem a jmenuje se vstup.txt. Stejným způsobem je možné zadat vstup ve variantě B i ve variantě A, v obou případech je nutné dbát na správnost zápisu.

V případě hraní ve variantě A se po zapsání otázky objeví na pravé straně její vyhodnocení. Před lomítkem je počet správně umístěných barev (bílé) a za lomítkem je počet správných barev, ale špatně umístěných (černé).

Pro případnou změnu počtu barev a počtu políček je nutné provést změnu v konstantách ve zdrojovém programu.

Výstupní data

Výstup z varianty D se nalézá ve vystup.txt. Na výstupu může být buď počet kroků, který programu stačilo k nalezení řešení, a nebo celý postup. V postupu se používá stejný zápis, jako při hraní ve variantě A.

Testovací data

Testovací data jsou přiložena. V jejich názvu je *n,k* pro která jsou použita. Použité omezení bylo 100. Jejich hlavní cíl je ukázat, že program doběhne v reálném čase. Jediná část programu, která se dá testovat jinak je v funkci kontrola. Její správnost lze ověřit výstupem s postupem. Dále byla odzkoušena hra s nalezením správného řešení i bez jeho nalezení.