

Zápočtový program zimní semestr 2020/2021

Hra Logik – Zuzana Bohatová

Program řeší hru Logik z hlediska hráče, který hádá. Pravidla hry: Hráč A si vybere nějakou 5místnou kombinaci z 8 barev (v kombinaci se barvy mohou opakovat), dále nazývanou jen jako kód. Hráč B se snaží kód uhodnout do 10 pokusů. Hráč A každý pokus ohodnotí kombinací černých a bílých kolíčků, kde černý znamená, že správná barva leží na správném místě, a bílý znamená, že barva v kódu je, ale není umístěná na správném místě.

V samotném programu jsou barvy nahrazeny čísly 0 až 7, místo černého kolíčku píšeme X a místo bílého O. Uživatel tedy zadá 5místnou číselnou kombinaci a program ji uhádne. Samotné vyhodnocení každého tahu neprovádí uživatel, ale funkce vyhodnocení.

Výběr algoritmu:

Na začátku jsem uvažovala mezi dvěma algoritmy, první byl začít odhady 01234 a 34567, a druhý postupně protestovat odhady (00011), (22233), (44455), (66677). Já jsem nakonec zvolila druhý zmíněný postup, jelikož u prvního zmiňovaného jsem měla z jejichž hodnocení vyvodíme, které kombinace barev může hádaný kód obsahovat. Tyto kombinace následně otestujeme, jestli by se délky vyhodnocení předchozích odhadů shodovaly, kdyby byla daná kombinace hledaným kódem. Tím získáme správná čísla.

Z těchto čísel poté vygenerujeme všechny možné permutace a postupně je procházíme a u každé otestujeme, jestli by předchozí odhady měly stejné hodnocení, kdyby tato permutace byla hledaný kód, v případě, že ano, tak se zeptáme je-li tato permutace hledaný kód, pokud ano, zvítězili jsme, pokud ne, pokračujeme otestováním další permutace, dokud nezvítězíme.

Program

Program začíná funkcí **spravnyvstup()**, která má podfunkci **nacticislo(m)**, která kontroluje, jestli uživatel zadal opravdu číslo ne nějaký jiný znak. Dále funkce **spravnyvstup()** kontroluje, jestli uživatel zadal opravdu jen čísla od 0 do 7, a jestli je čísel přesně pět, pokud jedna z těchto podmínek není splněná, napíše uživateli, kde je chyba a požádá ho, aby zadal čísla znovu. Zadanou kombinaci poté uloží do globální proměnné **kod**.

Další je funkce **Barvy(v, u, x, y, z)**, která nám postupně zadává odhady (00011), (22233), (44455), (66677), z kterých přibližně zjišťujeme, jaká čísla leží v hledaném kódu, společně s odhady máme proměnnou **počet**, která je na začátku 0 a postupně se k ní přičítá délka vyhodnocení každého z těchto čtyř odhadů. Každý odhad, který obsahuje alespoň jedno X či O, ukládáme, i s jeho vyhodnocením do slovníku *odhady*.

Z každého ohodnoceného odhadu se vytvoří pomocí funkce **kombsop(seznam, mist)** kombinace délky ohodnocení. A tuto kombinaci přidáme do listu *moznosti*. Např. když je hledaný kód je 01234, tak list *moznosti* vypadá takto: `[[[0, 0], [0, 1], [1, 1]], [[2, 2], [2, 3], [3, 3]], [[4], [5]]]`.

Funkce Barvy(v, u, x, y, z) skončí buď odhadem 66677 nebo ve chvíli, kdy se proměnná *počet*, do které se postupně přičítají délky ohodnocení jednotlivých tahů, rovná 5. Když už jsme vyzkoušeli odhad 66677 a počet se stále nerovná 5, tak víme, že je v něm buď 4 a více krát jedno z čísel na prvním místě odhadů (např. 0, 2, 4 či 6), nebo 3 a více krát jedno z čísel na posledním místě odhadů (např. 1, 3, 5 nebo 7). Tyto případy řešíme zvlášť. Pomocí `if` si rozdělíme funkci na tyto možnosti a „podmožnosti“:

1. počet se rovná 2 - hledaný kód je tedy 5krát číslo na posledním místě odhadu, který byl jediný ohodnocen např. 11111
2. počet se rovná 3 - zde máme dvě možnosti, máme ve slovníku *odhady*:
 - a. 1 odhad – zkusíme nejprve:
 - i. 5krát číslo na prvním místě v daném odhadu, např. 00000, a když vyhodnocení nebude mít délku 5, tak je jediná volba a to:
 - ii. 1krát číslo na 1. místě v odhadu a 4krát číslo na konci odhadu, např. 01111
 - b. 2 odhady – 4krát číslo na posledním místě odhadu, jehož ohodnocení má délku 2 a:
 - i. 1krát poslední číslo v odhadu, jehož ohodnocení má délku 1, např. 11113
 - ii. 1krát první číslo v odhadu, jehož ohodnocení má délku 1, např. 11112
3. počet se rovná 4 - zde máme více možností:
 - a. 3 odhady – jeden odhad A s ohodnocením délky 2 a dva odhady B, C s ohodnocením délky 1, začneme odhadem 1krát první číslo odhadu B, 1krát první číslo odhadu C a 3krát poslední číslo odhadu A, např. 11124, délku vyhodnocení tohoto odhadu nazveme *pom*, ta může mít hodnotu buď:
 - i. *pom* = 3: 3krát poslední číslo odhadu A, 1krát poslední číslo odhadu B a 1krát poslední číslo odhadu C, např. 11135
 - ii. *pom* = 4: 3krát poslední číslo odhadu A, 1krát první číslo odhadu B a 1krát poslední číslo odhadu C, pokud se vyhodnocení tohoto odhadu nerovná 5, např. 11125, tak už zbývá jediná možnost, a to 3krát poslední číslo odhadu A, 1krát poslední číslo odhadu B a 1krát první číslo odhadu C, např. 11134
 - iii. *pom* = 5: odhad je správný
 - b. 2 odhady
 - i. dva odhady s ohodnocením délky 2 - začneme odhadem 3x poslední číslo odhadu A a 2x poslední číslo odhadu B, např. 11133, tento typ vyhodnotíme a délku vyhodnocení si označíme jako proměnnou *pom*, ta se může rovnat buď:
 1. *pom* = 2: 2x první číslo odhadu A a 3x poslední číslo odhadu B, např. 00333
 2. *pom* = 3: 3x poslední číslo odhadu A a 2x první číslo odhadu B, např. 11122, pokud se délka vyhodnocení tohoto odhadu nerovná 5, tak už je jediná volba a to: 01333
 3. *pom* = 4: 2x poslední číslo odhadu A a 3x poslední číslo odhadu B, např. 11333, pokud se délka vyhodnocení tohoto odhadu nerovná 5, tak už je jediná volba a to: 3x poslední číslo odhadu A a 1krát každé z čísel odhadu B, např. 11132
 4. *pom* = 5: odhad byl správný
 - ii. odhad A délky 1 a odhad B délky 3 odhad – začneme s odhadem 1krát první číslo odhadu A a 4krát první číslo odhadu B, např. 02222, délku vyhodnocení tohoto odhadu označíme jako *pom*, ta se může rovnat buď:
 1. *pom* = 1: 3krát poslední číslo odhadu B, 1krát první číslo odhadu B a 1krát poslední číslo odhadu A, např. 33321
 2. *pom* = 2: 3krát poslední číslo odhadu B, 1krát první číslo odhadu A a 1krát první číslo odhadu B, např. 33302
 3. *pom* = 4: 4krát první číslo odhadu B a 1krát poslední číslo odhadu A, např. 22221
 4. *pom* = 5: odhad je správný
 - c. 1 odhad – jeden odhad s ohodnocením délky 4
 - i. 4krát první číslo a 1krát poslední číslo, např. 00001

ii. 3krát první číslo a 2krát poslední číslo, např. 00111

Poté jsou dvě možnosti buď je daná kombinace rovnou správná a program končí anebo jsou čísla správně ale ve špatném pořadí, v tom případě pokračujeme funkcí `Pozice()` (o té dále).

Dále máme proměnné a, b, c, d, e , což jsou listy s čísly 0 až 7, které reprezentují jednotlivá místa v kombinaci. U každého ohodnocení kontrolujeme, jestli nenastala jedna z těchto tří možností: ohodnocení obsahuje:

1. pouze samá O – provádí se u všech odhadů (přímo ve funkci `vyhodnoceni(odhad, proti)`), žádné z čísel není na správném místě, takže např. v případě odhadu *tuvxy*, se z listu *a* odstraní *t*, z listu *b* odstraní *u*, z listu *c* odstraní *v*, z listu *d* odstraní *x*, z listu *e* se odstraní *y*, tento krok se provádí u všech odhadů
2. X – provádí se jen u první čtyřech odhadů (00011 až 66677) přímo ve funkci `Barvy`, např. pro odhad *zzzww*, odstraní *w* z listu *a, b, c* a z listů *d, e*
3. XX – provádí se jen u první čtyřech odhadů přímo ve funkci `Barvy`, např. pro odhad *fffoo*, odstraní *f* z listu *d, e*

Další funkcí je funkce **vyhodnoceni(odhad, proti)**, která nám porovnává proměnnou odhad s proměnnou proti – a vypíše daná X a O, v případě, že je odhad a proti totožné a proti se rovná kódu, tak vypíše „Vyhráli jste“ a vítěznou kombinaci a program ukončí, v případě, že se délka vyhodnocení rovná 5, ale kombinace není správná přidá tuto kombinaci do proměnné *spravnebarvy*.

Dále pokračujeme funkcí **KombiBarev()**, kde sestavíme všechny možné 5místné kombinace z listu *moznosti*, např. 00224, 00225, 00234 ... a uložíme je do listu *kombinace*. Vytvoříme list *mista* z listů *a, b, c, d, e* vytvoříme kombinace, které obsahují vždy po jednom prvku z *a, b, c, d, e*. Tyto kombinace přidáváme do listu *mista*. List *mista* poté porovnáme s listem *kombinace* a ty prvky které mají stejné vložíme do listu *shoda*.

Pokračujeme funkcí **NajdiBarvy(shoda)** - projde postupně všechny prvky listu *shoda* a zjistí, jestli čísla daného prvku *x* mohou být hledaná čísla (porovná délku hodnocení u jednotlivých odhadů uložených ve slovníku *odhady* s délkou ohodnocení stejných odhadů, kdyby kód byl prvek *x*). V případě, že prvek *x* může být hledaným kódem, tak použijeme funkci `Pozice()`.

Funkce **Pozice(spravnebarvy)** vytvoří permutace z čísel, která obsahuje list *spravnebarvy*, z každého ohodnoceného odhadu vytvoří kombinace délky ohodnocení. A tuto kombinaci přidáme do listu *moznosti*. Např. když je hledaný kód je 01234, tak list *moznosti* vypadá takto: `[[[0, 0], [0, 1], [1, 1]], [[2, 2], [2, 3], [3, 3]], [[4], [5]]]`. Dále pokračujeme funkcí `KombiBarev()`, kde sestavíme všechny možné 5místné kombinace z listu *moznosti*, např. 00224, 00225, 00234 ... a uložíme je do listu *kombinace*.

Reprezentace vstupních dat a jejich příprava

Vstupní data jsou pětice čísel skládající se z 0, 1, 2, 3, 4, 5, 6, 7. V případě, že uživatel zadá více nebo méně čísel nebo mezi čísly bude 8 nebo 9, program napíše, kde se stala chyba a požádá o nová vstupní data. V případě, že vstup není číslo, program vypíše, že jste nezadali číslo a opět požádá o nový vstup.

Reprezentace výstupních dat a jejich interpretace

Program vypisuje postupně každý odhad *uvxyz* a jeho vyhodnocení, ve formátu: `[u, v,`

x, y, z] a ['X', 'X', 'X', 'O', 'O']. Ve chvíli, kdy uhádnete kód, tak program vypíše např. „Vyhrál jsem na 7 . pokus, správná kombinace je: [7, 0, 4, 3, 1]“

Průběh prací

Práce probíhala na začátku poměrně těžko, je to můj první takto rozsáhlý program, takže mi chvíli trvalo vychytat správný postup. Vypracovávala jsem postupně jednu funkci po druhé, začala jsem kontrolou vstupu a poté vyhodnocením kombinace, jednotlivé funkce jsem postupně upravovala a přidávala jim některé úkoly na víc, které mi pomáhaly v zúžení možných kombinací. S výsledkem jsem poměrně spokojená, ale myslím, že by se to nejspíše dalo udělat jednodušeji.

Sada testovacích dat

Pro každý if ve funkci Barvy – jeden zkušební kód: 22222, 11111, 23333, 11311, 33336, 11124, 11135, 11125, 77734, 66667, 22221, 02333, 21333, 02222, 11132, 11333, 67333, 33366, 66555, 11331

A několik dalších zkušebních kódů: 01234, 43210, 71610, 52710, 61443, 44215, 14624, 51510, 17350.