KSI 2012
Úloha 2-1: Lovci

Jan Horáček
Gymnázium, Brno, Vídeňská 47; jan.horacek@seznam.cz

11. listopadu 2012

1 Úvod

Problémem těchto úloh "někdo umí pouze něco a nic víc"bývá stanovení hranic toho, co opravdu umí. Je sice hezké, že můžeme gumovat a zapisovat na vygumovaná místa, ale co si umí pravěccí lidé pamatovat? Resp. kolik si toho umí pamatovat? A jaký mají "výpočetní výkon"?

Všechny mé snahy vedly k vytvoření takového algoritmu, který by byl nejsnažší, ba i negramotní lidé by ho mohli zvládnout. A jelikož tématém této sady jest třídění, řešení bude asi mít něco společného s tříděním.

Základní prozření nastalo ve chvíli, kdy jsem si uvědomil, že pro zíkání nejúspěšnějšího dne nepotřebujeme znát počet ulovených tygrů, nýbrž nám stačí pouze setřídit tabulku podle počtu chycených tygrů.

2 Zdrojový kód

K tomuto algoritmu není přiložen žádný spustitelný kód, pouze zde bude slovně popsán algoritmus, kterým lze daný problém jednodušše vyřešit.

Tento popis řešení obsahuje úryvky preudokódu. Tento preudokód je založen na jazyce C.

3 Popis řešení

Hlavní část postupu tvoří setřídění křížků a koleček na každém řádku tak, aby byly všechny křížky vpravo a všechna kolečka vlevo.

Následuje kompetní slovní popis funkce programu.

3.1 Bubble sort

A toto tříděni bude nejjednodušší provádět Bubble sortem, tedy tříděním probubláváním.

Návod pro pravěkké lidi:

Následující postup proveďte pro každý řádek samostatně:

Nasměrujte veškerou svou koncentraci na první 2 znaky zleva v prvním řádku. Pokud se vlevo nachází křížek (nic) a vpravo kolečko (tygr), tak tyto dva znaky prohoďte. Pak se přesuňte o jeden znak doprava a prostup provádětejte, dokud nebude celý řádek setříděn. To poznáte tak, že jste na celém řádku nemuseli prohodit žádné 2 prvky.

Zasvěceným informatikům je jasné, že se jedná o Bubble sort aplikovaný na každý řádek samostatně. Výsledná vzorová tabulka by tak měla po tomto postupu vypadat následovně:

O O O X X
O X X X X
O O X X X
O O O X X
O O O O X
O O O X X
O O O X X

Pseudokód k této části postupu by pak mohl vypadat následovně:

```
for (radek=0;radek<pocet_radku;radek++)
{
 do
 {
  setrideno = true;
  for (sloupec=0;sloupec<(pocet_sloupcu-1);sloupec++)
   if ((tabulka[sloupec,radek] == 'X') &&
       (tabulka[sloupec+1,radek] == 'O'))
    {
     //prohodime 2 prvky
     setrideno = false;
    }
 while (!setrideno)
}
```

Listing 1: Bubble sort

3.2 Nalezení nejúspěšnějšího dne

S takto upravenou tabulkou už uděláme krátký proces:

Čtěte postupně sloupce počínaje pravým sloupcem nahoře, konče levým sloupcem dole. Tedy, po přečtení jednoho sloupce se posuňte o sloupec doleva a čtěte ho opět zezhora. Jakmile narazíte na znak "O", našli jste nejúspěšnější den. Pořadí řádků se nezměnilo, takže můžete rovnou vyhlásit tento den jako nejúspěšnější.

4 Závěr

Tento algoritmus se potýká s jedním zásadním problémem: nalezne pouze první z množiny nejúspěšnějších dnů a to takový den, který byl v týdnu nejdříve. Pro nalezení všech nejúspěšnějších dnů by bylo potřeba projít celý sloupec, ve kterém jsme našli ono první

"O". Pokud bychom v takovémto sloupci našli další "O", je den na tomto řádku také jeden z nejúspěšnějších.

Z hlediska prostoru si klade tento postup na pravěkké lidi relativně malé požadavky: umět se orientovat v tabulce a pamatovat si, jestli prohodili nějaké 2 prvky.

Pevně věřím, že po několika opakováních se tento postup stane stejně mechanickým, jako lovení šavlozubých tygrů.

Reference

Tento algoritmus byl vytvořen pouze autorovou vlastní silou - bez použití externích zdrojů.