

Jméno: Karel Kubíček

UČO: 408351

Jméno: Henrich Lauko

UČO: 410438

Algoritmus pro zpřehlednění pracuje s globální 3D maticí *Matrix*, která má rozměry $\frac{n}{3} + 1 \times \frac{n}{3} + 1 \times \frac{n}{3} + 1$. Každé políčko matice obsahuje hodnotu maximální ceny z nákupu prvních x aut v prvním autosaloně, y v druhém a z ve třetím autosaloně. Kromě toho také obsahuje flag, který popisuje, z kterého směru jsme se na danou pozici dostali. Za okraji této matice se nachází hodnoty 0, obdobně se nula nachází v $C[0][i]$.

Procedura OPTIMALPRIZE(C)

```

vstup:  $C$ , je matice vyjadřující cenu
výstup: uspořádaná  $n$ -tice, popisující které auta kam prodat za maximální cenu
1  $Matrix[0][0][0] \leftarrow 0$  // 0 aut prodáme za cenu 0
2 for  $x \leftarrow 0$  to  $\frac{n}{3}$  do
3   for  $y \leftarrow 0$  to  $\frac{n}{3}$  do
4     for  $z \leftarrow 0$  to  $\frac{n}{3}$  do
5        $prizeX \leftarrow Matrix[x-1][y][z] + C[x+y+z][1]$ 
6        $prizeY \leftarrow Matrix[x][y-1][z] + C[x+y+z][2]$ 
7        $prizeZ \leftarrow Matrix[x][y][z-1] + C[x+y+z][3]$ 
8        $Matrix[x][y][z] \leftarrow \text{MAX}(prizeX, prizeY, prizeZ)$  // MAX vrátí kromě
        maxima i flag odpovídající směru, ze kterého bylo maximum vybráno
9     od
10   od
11 od
12  $Output \leftarrow$  prázdné pole délky  $n$ 
13  $car \leftarrow Matrix[\frac{n}{3}][\frac{n}{3}][\frac{n}{3}]$ 
14 for  $i \leftarrow n$  downto 1 do
15    $Output[i] \leftarrow car.flag$  // flag určuje který autosalón jsme pro auto použili
16    $car \leftarrow car.prev()$  // na základě flagu vrátí danou pozici v Matrix
17 od
18 return  $Output$ 

```

Popis algoritmu: algoritmus v každé iteraci spočítá jedno políčko matice *Matrix*, přičemž to je vybráno jako maximum okolních hodnot plus auto které aktuálně prodáváme v jakém autosalonu, což odpovídá směru, ve kterém jsme se v matici posunuli (x je první autosalón, y druhý a z třetí). Využili jsme memoizace, což je prostředek dynamického programování, abychom nemuseli vypočtené hodnoty opakovaně počítat.

For cyklus počínající na 14. řádku nám pomocí uložených flagů nalezne, kudy k výslednému řešení vedla cesta.

Funkce *prev* na základě flagu vrací přecházející prvek v matici podle směru x , y , z (pro flag x vrací ukazatel na prvek $Matrix[x-1][y][z]$, obdobně pro ostatní flagy).

Korektnost: algoritmus je totálně korektní, pokud je konečný a parciálně korektní, což znamená výstup splňuje výstupní podmínku, kterou je v našem případě, že nalezená n -tice odpovídá řešení s maximální cenou, za kterou lze auta prodat. n -tice musí obsahovat od každého autosalonu stejný počet aut, což je zajištěno rozměry matice. Konečnost algoritmu je zřejmá, jelikož používáme jen for cykly a hodnoty v nich jen inkrementujeme.

Jméno: Karel Kubíček

UČO: 408351

Jméno: Henrich Lauko

UČO: 410438

Parciální korektnost lze dokázat indukcí. Pro 0 aut vybereme cenu 0. Induktivním předpokladem je, že pro k aut vybereme cenu p , která je nejlepší možná. Při přidání dalšího auta dostaneme 2 různé možnosti.

1. původní maximální cesta je součástí nové maximální cesty. Pak jen auto zařadíme do autosalónu, který nabízí největší cenu za nové auto.
2. původní maximální cena není součástí správného řešení pro $k + 1$ aut. Pak musíme někde v průběhu výpočtu přejít do o jedna větší souřadnice, než byl rozsah původního řešení. Pak už v této souřadnici nesmíme dále pokračovat, abychom neporušili rovnoměrnou distribuci.

Rozdíl počtu prodaných aut do autosalónu o 1 nepovažujeme za porušení rovnoměrnosti distribuce.

Jelikož je algoritmus konečný a parciálně korektní, pak je nutně i korektní.

Asymptotická časová složitost je $\in \mathcal{O}(n^3)$, for cykly mají daný rozsah $(\frac{n}{3})$ a indexy se každým průchodem zvyšují o 1. Vyhodnocení maxima je provedeno v konstantním čase. Závěrečný for cyklus má jen lineární složitost, asymptotickou složitost tedy nemění.