

Meno: Tomáš Belan

Príklad 6, kat. O

Škola: ŠpMNDaG Teplická, Bratislava

Strana 1 z 1

Toto riešenie má maximálnu časovú náročnosť $O(N^2)$. Áno, nie je to ktoviečo, a asi by to išlo aj lepšie, ale neviem, že ako, tak pošlem aspoň toto.

Zoberme si príklad zo zadania. Nie je vhodné dať cenu výrobku 8: výrobok si kúpia tí istí, ako keby mal cenu 9 (nikto nemá 8 ako maximálnu cenu), takže zisky budú len menšie (na každom, čo si výrobok kúpil, ešte oproti cene 9 stratujú 1 peniaz). Toto pravidlo sa dá zovšeobecniť: dávať cenu výrobku, ktorá nie je nikoho maximálnou cenou, je neoptimálne.

Keď spravím túto optimalizáciu, zvyšok je pomerne základný. V hash tabuľke (jej popis vid' wikipedia; časová náročnosť je „v podstate“ $O(1)$) mám množinu maximálnych cien jednotlivých zákazníkov. (Množina tu znamená, že používam len kľúče hash tabuľky, všetky hodnoty sú proste 'true'.) Pre každú cenu spočítam v cykle $O(N)$, aký je profit. Ak je lepší ako doterajšie maximum, použije sa.

Tu je moja implementácia popísaného algoritmu v Ruby.

```
print "Zadaj N: "; N = readline.to_i
riadky = Array.new
max_ceny = Hash.new
N.times do |i|
  print "Zadaj riadok: "
  riadok = readline.split.collect{ |slovo| slovo.to_i }
  # ak a <= b, tak si to nikdy nekupi; odignorujme ho
  next if riadok[0] <= riadok[1] # next je ako continue v C
  riadky << riadok
  # pridam max.cenu tohto zakaznika do mnoziny max_ceny
  max_ceny[riadok[0]] = true
end

max_profit_cena = 0; max_profit = 0

max_ceny.each do |cena,tmp|
  profit = 0 # ideme vyratat profit pri tejto cene
  riadky.each do |riadok|
    profit += cena - riadok[1] if riadok[1] < cena and cena <= riadok[0]
  end
  if profit > max_profit
    max_profit = profit; max_profit_cena = cena
  end
end

puts max_profit_cena
```