

Cvičenia č. 4

2-AIN-105, Zima 2013

5. novembra 2013

1. Máme policu s n vínami. Každý rok predáme práve jedno z dvoch vín, ktoré sú aktuálne na okraji police. Pre každé víno vieme jeho cenu ak by sme ho predali prvý rok: $A[1], A[2], \dots, A[n]$ a zároveň vieme, že ak i -te víno predáme x -ty rok, tak zaňho dostaneme $xA[i]$. Vašou úlohou je zistiť koľko najviac vieme na predaji zarobiť.

Riešenie: Greedy algoritmus, ktorý každý rok predá lacnejšie víno nefunguje. Protipríklad vidno na vstupe: 1010111, kde greedy algoritmus zarobí $10 + 20 + 3 + 44 = 77$ a optimálne riešenie zarobí $11 + 2 + 30 + 40 = 83$.

Správne riešenie je použiť dynamické programovanie. Podproblémom bude $B[i, j]$ – koľko najviac môžeme zarobiť na vínach medzi pozíciami i, j . Hodnotu $B[i, j]$ nám vie spočítať nasledujúci program:

```
rok = n - j + i
if i == j:
    B[i, j] = rok * A[i]
else:
    B[i, j] = max(rok * A[i] + B[i+1, j], rok * A[j] + B[i, j-1])
```

Vždy máme na výber z dvoch možností, buď vezmeme fľašu zľava alebo sprava. Zbytok máme vyriešený v menšom podprobléme. Tento program už len vhodné obalíme, aby sme vypočítali každú potrebnú hodnotu. Treba si dať pozor, aby sme hodnoty počítali v správnom poradí. Nemôžeme len naivne urobiť dva forcykly cez i, j , lebo pri počítaní $B[i, j]$ potrebujeme $B[i + 1, j]$. Správne je počítať najprv podproblémy dĺžky 1, potom dĺžky 2, Alebo použiť rekúziu s memoizáciou, ktorá tento problém rieši za nás. Každopádne dostávame riešenie s časovou a pamäťovou zložitostou $O(n^2)$.

2. Máme text, ktorý pozostáva z n slov. Vieme dĺžku každého slova $W[1], W[2], \dots, W[n]$. Tieto slova chceme rozdeliť do riadkov, tak aby na každom riadku bolo maximálne k znakov. Pri rozdeľovaní chceme minimalizovať pokutu, ktorá je definovaná nasledovne: Celková pokuta je súčet pokút za jednotlivé riadky. Pokuta za riadok je: $k - s_i$, kde s_i je celkový počet znakov (bez mezdier) na riadku i . Nájdite jedno rozdelenie slov do riadkov, ktoré minimalizuje pokutu.

Riešenie: V tomto prípade funguje greedy algoritmus. Vkladáme do riadku toľko slov koľko sa tam zmestí a potom sa presunieme na ďalší riadok. Optimálnosť greedy algoritmu ukážeme nasledovnou lemov: Nech $r_1, r_2, \dots, r_n$ sú čísla riadkov, na ktorých sú slová v našom riešení. Nech $o_1, o_2, \dots, o_n$ sú čísla riadkov, na ktorých sú slová v nejakom optimálnom riešení. Ukážeme, že existuje optimálne riešenie, ktoré sa zhoduje s našim. Dôkaz ako obvykle indukciou, kde indukčný krok je nasledovný: Nech sa riešenia zhodujú na prvých k miestach, potom existuje optimálne riešenie, ktoré sa zhoduje s našim na $k + 1$ miestach. Toto ukážeme tak, že $o_{k+1} \neq r_{k+1}$, tak musí platiť $o_{k+1} = r_{k+1} + 1$. Ale keď $k + 1$ slovo presunieme o riadok vyššie, tak sa pokuta nezmení a teda máme optimálne riešenie, ktoré sa zhoduje s našim na $k + 1$ slovách.

Rozmyslite si ešte zvyšné detaily dôkazu. Časová zložitosť algoritmu bude $O(n)$.

3. Máme text, ktorý pozostáva z n slov. Vieme dĺžku každého slova $W[1], W[2], \dots, W[n]$. Tieto slova chceme rozdeliť do riadkov, tak aby na každom riadku bolo maximálne k znakov. Pri rozdeľovaní chceme minimalizovať pokutu, ktorá je definovaná nasledovne: Celková pokuta je súčet pokút za jednotlivé riadky. Pokuta za riadok je: $(k - s_i)^2$, kde s_i je celkový počet znakov (bez mezdier) na riadku i . Nájdite jedno rozdelenie slov do riadkov, ktoré minimalizuje pokutu.

Riešenie: Toto už nie je riešiteľné greedy algoritmom. Problém je, že presunutie slova na iný riadok nám už môže zmeniť celkovú pokutu (narozdiel od predchádzajúcej úlohy). Dobré riešenie je opäť dynamické programovanie, kde podproblém je $B[i]$ – akú najmenšiu pokutu vieme dosiahnuť, ak by sme mali iba prvých i slov. Toto vieme z predchádzajúcich podproblémov vypočítať tak, že vyskúšame všetky možnosti na dĺžku posledného riadku. Rozmyslite si ako to urobiť a že výsledná časová zložitosť bude $O(nk)$.

4. Máme zadaný strom, ktorý má koreň vo vrchole 0. Vašou úlohou je vybrať z neho čo najviac hrán tak, aby žiadne dve nemali spoločný vrchol.

Riešenie: Sú dve možnosti.

Dynamické programovanie – pre každý podstrom si spočítame koľko hrán vieme z neho vybrať.

Greedy – vyberieme si list, hranu idúcu z neho a vymažeme zo stromu všetky hrany, ktoré sa danou hranou susedia.

Dôkaz greedy algoritmu bude v tomto prípade mierne atypický. Dokážeme nasledujúce tvrdenie: Pre každý les veľkosti n je náš algoritmus optimálny (les je graf, ktorý pozostáva z jedného alebo viacerých stromov, inými slovami acyklický graf, ktorý nie je nutne súvislý). Budeme robiť indukciu vzhľadom na n . Prípady s malým n nechávame na čitateľa. Indukčný krok: Nech tvrdenie platí pre $n \leq k$, ukážeme, že platí aj pre $n = k + 1$. Náš algoritmus vyberie list u , ktorý je spojený s vrcholom v . Ak optimálne riešenie nepoužije hranu (u, v) , potom vrchol v , musí byť spojený s iným vrcholom w . Ale nič nepokazíme ak miesto hrany (v, w) použijeme hranu (u, v) . Dôkaz hotový.

Každopádne aj dynamické programovanie aj greedy algoritmus vieme pri šikovnej implementácii implementovať v čase $O(n)$.