

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE INFORMÁTICA

PROBLEMA 1310

Lucro

JOÃO HENRIQUE ZANATA DE CARVALHO RA: 95687

VANESSA YUKARI KAJIHARA RA: 78605

Problema

- ❖ O dono de um circo sabe o quanto de receita ele pode obter em qualquer dia de uma série de dias em uma cidade.

$$R = \{r_1, r_2, \dots, r_n\}$$

- ❖ Ele também sabe o custo C constante diário para manter o circo.
- ❖ Define-se $S = \{a_1, a_2, \dots, a_n\}$ o lucro de cada dia.
- ❖ O objetivo é calcular o lucro máximo.

Formulação recursiva

$$opt(n) = \begin{cases} S(n) & se\ n = 0 \\ \max \{opt(n - 1) + S(n), S(n)\} & se\ n > 0 \end{cases}$$

Subestrutura ótima

Queremos determinar um intervalo máximo de lucro, $S_{ij} = \{a_i, \dots, a_j\}$, que inclui algum lucro a_k

❖ Se $a_k \in S_{ij}$, então obtemos dois subproblemas

$S_1^* = \{a_i, \dots, a_{k-1}\}$ e $S_2^* = \{a_{k+1}, \dots, a_j\}$ e a subsequência com soma máxima em S_{ij} é a soma dos elementos de S_1^* , S_2^* e a_k .

Sobreposição de problemas

$$\{-2, 15, -14, 60, -5, 1\}$$

$$\text{Soma 1: } -2 = -2$$

$$\text{Soma 2: } -2 + 15 = 13$$

$$\text{Soma 3: } -2 + 15 - 14 = -3$$

$$\text{Soma 4: } -2 + 15 - 14 - 3 + 60 = 57$$

Sem programação dinâmica

maxSumSubArray(A, n)

$max \leftarrow -1$

for $i \leftarrow 0$ to n do

$soma \leftarrow 0$

for $j \leftarrow 0$ to n do

$soma \leftarrow soma + A[j]$

if $soma > max$ then

$max \leftarrow soma$

Com programação dinâmica

maxSumSubArray(A, n)

$local_max \leftarrow 0$

$global_max \leftarrow -\infty$

for $i \leftarrow 0$ **to** n **do**

$local_max \leftarrow \max\{A[i], A[i] + local_max\}$

if $local_max > global_max$ **then**

$global_max = local_max$

return $global_max$

Complexidade?

Sem PD: $O(n^2)$

Com PD: $O(n)$

Casos de teste

Caso 1:

6

20

18 35 6 80 15 21

Kadane: 61

Força bruta: 61

Casos de teste

Caso 2:

4

40

30 20 10 38

Kadane: 0

Força bruta: 0

Casos de teste

Caso 3:

3

10

12 13 100

Kadane: 95

Força bruta: 95

Casos de teste

Caso 4:

25

90

87 28 3 783 7 43 4 8 78 25 37 600 457 0 14 578 157 378 124 1 34

78 95 45 45

Kadane: 1853

Força bruta: 1853

Casos de teste

Caso 5:

5

1

2 3 4 5 6

Kadane: 15

Força bruta: 15

Escolha gulosa

- ❖ Em uma etapa de pré-processamento criar um vetor auxiliar que contém as somas de todos os valores até então;
- ❖ Escolher o maior valor deste vetor auxiliar, ele é o limite final do subvetor que resulta na soma máxima;
- ❖ Faça a mesma coisa, porém iniciando no limite final até a primeira posição do vetor;
- ❖ O maior valor desse novo vetor é a soma máxima possível para esse problema;

Exemplo:

Vetor: $\{-2, 15, -14, 60, -5, 1\}$

Somando todos os valores iterativamente:

VetorAux: $\{-2, 13, -1, \mathbf{59}, 54, 55\}$

Repetindo o processo ao contrario temos:

VetorAux $\{59, \mathbf{61}, 46, 60, \dots\}$

Sendo assim 61 é a maior soma possível para esse problema