

consumo

A linha 2 consome  $O(n-k+1)$  com  $k$  constante, podemos considerar  $O(n)$ . O algoritmo utiliza o algoritmo select BEPRT que recebe um vetor de  $k$  elementos e o  $q$ -ésimo menor elemento. Seu consumo  $O(k)$ .

A linha 3+b consome  $O(k)$ . Portanto o algoritmo possui consumo  $O(n \cdot k)$ .

b) Escreva um algoritmo Mediana 2 ( $A, n, k$ ) que recebe um vetor de  $n$  inteiros  $A[1..n]$  e  $k$  e devolve um vetor  $B[1..n-k+1]$  das medianas dos  $k$ -segmentos de  $A$ . Seu algoritmo deve ter tempo  $O(n \lg k)$ . Explique pq ta certo e tem o consumo. Pode usar árvore balanceada.

Mediana 2 ( $A, n, k$ )

|   |                                   |                       |
|---|-----------------------------------|-----------------------|
| 1 | TREE-BUILD ( $T, A, 1, k-1$ )     | $O(k \lg k)$          |
| 2 | Para $i \leftarrow 1$ até $n-k+1$ | $O(n)$                |
|   | TREE-INSERT ( $T, A[i+k-1]$ )     | $O(\lg k) \cdot O(n)$ |
|   | $B[i] \leftarrow T[0]$            | $O(n) \cdot O(n)$     |
|   | TREE-DELETE ( $T, A[i]$ )         | $O(\lg k) \cdot O(n)$ |
|   | devolva $B$                       | $O(1)$                |

consumo

$O(n \lg k)$

Parâmetros

Inicia-se a árvore com  $k-1$  elementos. Para cada  $i$  no  $n$  estabelece a árvore  $T[i..i+k-1]$  e põe a sua raiz que representa a mediana, e após isso retira-se o elemento  $A[i]$  permanecendo com  $T[i..i+k-2]$  depois na linha 2 para que seja verificada a próxima elemento e devolva o 1°.

$V[i] = [i] - 1$

189

PROVA 2004/1

Questão 1 - pg 138

Questão 2 - Meu manual diz que um problema "está em NP" denota que não existe algoritmo polinomial para o problema. Está certo ou errado? Justifique.

Errado. Pois  $P \subseteq NP$  dessa forma existe um subconjunto em NP onde os problemas são resolvidos em tempo polinomial.

Questão 3 - FATOR ( $n$ )

|   |                                 |
|---|---------------------------------|
| 1 | Para $d \leftarrow 2$ até $n-1$ |
| 2 | se resto ( $n/d$ ) = 0          |
| 3 | devolva $d$                     |
| 4 | devolva 0                       |

Este algoritmo não é polinomial. Está certo?

$n$  = de dígitos  $\lg(n+1)$

Tamanho da instância:  $(\lg(n+1))$  e  $n = 2$

linha consumo  $O(2^{\lg(n+1)})$

4  $O(1)$   $\lg(n+1)$

$T(\lg(n+1)) = 2$

Está certo

Seja o tamanho da entrada  $m$  (n de dígitos), então o consumo de tempo para  $T(n) = O(n)$  mas  $m = \lg n \Rightarrow 2^m = 2^{\lg n} \Rightarrow \sqrt{2^m} = 2$

Questão 4 -  $|V[i] - V[i+1]| \leq 1$  - BUSCA ( $V, n, z$ ) -  $O(\lg n)$

Ex: 1 2 3 4 3 4 5 4 5 6  $k$ -ordenado

BUSCA ( $V, p, n, z$ )

se  $p = n+1$   
se  $A[p] = z$   
devolva  $n$   
senão devolva -1

senão  $q \leftarrow \lfloor (p+n)/2 \rfloor$

se  $A[q] < z$   
então devolva BUSCA ( $V, q+1, n, z$ )

senão devolva BUSCA ( $V, p, q-1, z$ )



$$q(n) = O\left(T\left(\frac{n}{2}\right) + 1\right)$$

$$T(n) = 0 \quad \text{para } n=1$$

$$T(n) = T\left(\frac{n}{2}\right) + 1 \quad \text{para } n \geq 2^1, 2^2, \dots$$

$$= T\left(\frac{n}{4}\right) + 1 + 1$$

$$= T\left(\frac{n}{8}\right) + 1 + 1 + 1$$

$$= T\left(\frac{n}{2^i}\right) + 1 + 1 + \dots + 1$$

$$= T(0) + \log n$$

$$= \log n$$

Vou provar que  $T\left(\frac{n}{2}\right) + 1 \leq \log n$  para  $n \geq 1$

Prova por indução em  $n$

base:  $n=1$

$$T(0) = 1 = \log 1 = 0$$

para  $n \geq 2^1, \dots$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$$\stackrel{H.I.}{\leq} \log \frac{n}{2} + 1$$

$$= \log n - \log 2 + 1$$

$$= \log n - 1 + 1$$

$$= \log n //$$

Conclusão:

o vetor  $V$  não está totalmente ordenado, mas é crescente, então aplica-se a busca binária (já conhecida que possui consumo  $O(\log n)$ ).

Questão 5 - pg 128

$$n=1 \quad T(n)=0$$

$$n=2 \quad T(n)=n^3$$

$$n=3 \quad T(n)=n^3$$

se  $n=1$  então desce para  $n=4$   $T(n)=8 \cdot T\left(\frac{n}{2}\right) + n^3$

para  $i=1$  até 8 (caso  $Z \leftarrow \text{Algo}(m/2)$ )

para  $i=1$  até  $n^3$  (caso  $Z \leftarrow 0$ )

a)  $T(n) = 0 \quad n=1$

$$T(n) = 8T\left(\frac{n}{2}\right) + n^3 \quad n \geq 2^1, 2^2, \dots$$

b)  $T(n) = 8T\left(\frac{n}{2}\right) + n^3$

$$= 8 \cdot \left(8T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^3\right) + n^3$$

$$= 8^2 \left(8T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^3\right) + n^3 + n^3$$

$$= 8^i \left(T\left(\frac{n}{2^i}\right) + \left(\frac{n}{2^i}\right)^3\right) + n^3 \cdot i$$

$$= 8^{\lg n} \cdot T(1) + n^3 \cdot \lg n$$

$$= 0 + n^3 \cdot \lg n = n^3 \lg n$$

$$2^3 \lg 2 \quad 3^3 \lg 3$$

Solução fechada =  $n^3 \lg n$

Vou provar que a solução está correta  $T(n) = n^3 \lg n$

Prova por indução em  $n$ :

base:  $n=1$

$$T(1) = 0 = 1^3 \lg 1 = 0 //$$

para:  $n \geq 2^1, 2^2, \dots$

$$T(n) = 8T\left(\frac{n}{2}\right) + n^3$$

$$\stackrel{H.I.}{\leq} 8 \left( \left(\frac{n}{2}\right)^3 \lg \frac{n}{2} \right) + n^3$$

$$= n^3 (\lg n - 1) + n^3$$

$$= n^3 \lg n - n^3 + n^3$$

$$= n^3 \lg n //$$

$$\text{Portanto } T(n) \geq \frac{1}{2} n^3 \lg n$$

$$\frac{n^3 \lg n}{2} \geq \frac{1}{2} n^3 \lg n \quad \text{para } n \geq 1$$

$$\text{Assim } T(n) = \Omega(n^3 \lg n) //$$



e) troque 8 por 4

$$q(n) = 0 \quad n=1$$

$$q(n) = 4 \left( t(n/2) \right) + n^3 \quad n \geq 2^1, 2^2$$

$$t(n) = 4 \left( t(n/2) \right) + n^3$$

$$= 4 \left( 4 \left( t(n/4) \right) + \left( \frac{n}{2} \right)^3 \right) + n^3$$

$$= 4^2 t(n/4) + 4 \cdot \left( \frac{n}{2} \right)^3 + n^3$$

$$= 4^2 \left( 4 \left( t(n/8) \right) + \left( \frac{n}{4} \right)^3 \right) + 4 \left( \frac{n}{2} \right)^3 + n^3$$

$$= 4^3 t(n/8) + 4^2 \left( \frac{n}{4} \right)^3 + 4 \left( \frac{n}{2} \right)^3 + n^3$$

$$= 4^i t(n/2^i) + \sum_{k=0}^{i-1} \left( \frac{4}{2^3} \right)^k n^3$$

$$= 4^{\lg n} t(1) + n^3 \sum_{k=0}^{\lg n - 1} \left( \frac{4}{2^3} \right)^k$$

$$= 0 + n^3 \cdot \left( \frac{\frac{4}{2^3} - 1}{\frac{4}{2^3} - 1} \right)$$

$$= n^3 \cdot \left( \frac{4^{\lg n} - 1}{2^{\lg n} - 1} \right) \cdot (-2^3)$$

$$= n^3 \cdot \left( \frac{4^{\lg n} - 1}{2^{\lg n} - 1} \right) \cdot (-2^3)$$

$$= n^3 \cdot \left( \frac{4^{\lg n} - 1}{n^3} \right) \cdot (-2^3)$$

$$= (4^{\lg n} - n^3) \cdot (-2^3)$$

$$= 8n^3 - 8 \cdot 4^{\lg n}$$

$$\text{Fórmula fechada} = 8n^3 - 8 \cdot 4^{\lg n}$$

ou provar que esta solução está correta.

Prova por indução em  $n$ :

base:  $n=1$

$$t(1) = 0 = 8 \cdot 1^3 - 8 \cdot 4^{\lg 1} = 8 - 8 = 0$$

passo:  $n \geq 2^1, 2^2, \dots$

$$t(n) = 4 \left( t(n/2) \right) + n^3$$

$$\stackrel{H.I.}{\leq} 4 \left( 8 \cdot \left( \frac{n}{2} \right)^3 - 8 \cdot 4^{\lg \frac{n}{2}} \right) + n^3$$

$$= 4 \cdot \left( n^3 - 8 \cdot 4^{\lg n - 1} \right) + n^3$$

$$= 4n^3 - 8 \cdot 4 \cdot 4^{\lg n - 1} + n^3$$

$$= 8n^3 - 8 \cdot 4^{\lg n}$$

$$= t(n)$$

Assim podemos provar que  $t(n) \leq 8n^3$  para  $n > 1$

$8n^3 - 8 \cdot 4^{\lg n} \leq 8n^3$  para  $n > 1$  comprovadamente

é provado.

$$t(n) = O(n^3) //$$

Questão 8 - FAZ-ALGO(E, n)

$t \leftarrow 1, 2, 3$

$n=5$

1º 0

$t \leftarrow 0, 1, 2$

2º 0

E 

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 1 | 1 | 1 |

Para E todos iguais não executa nunca

S 

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|

E 

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 1 | 1 | 1 |

$j = 1 \times 3$

1º 0

$t = 0, 1, 0, 1, 0$

2º 1

S 

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|

E 

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 1 | 1 | 1 |

$j = 1 \times 2 \times 4 \times 5$

$t = 0, 1, 0, 1, 2, 3, 0$

O Aluno está certo. A linha 4 executa n vezes tendo um tempo de  $O(n)$ , mas apesar de existir na linha 5 um loop interno este executa no máximo 1 vez a cada iteração. As condições para entrar no loop são  $t > 0$  e os elementos de E serem diferentes. Se considerarmos E igual nunca vai executar, se considerarmos todos diferentes o loop vai ser executado apenas 1 vez pois  $t = 5$  que é zero, ou seja, variável entre 1 e 0. E se obtivermos intercalação de valores iguais ele também não passará de uma execução pois quando t mudar de valor o valor de E não muda. O loop depende de uma combinação de E diferente e não passaria novamente pelo loop.



Entrar a linha 5, vai para  $O(1)$   
 dando como retorno data e  $O(m)$ .

Questão 6 - Subsequência - SUBSEQ-CONT( $X, m, Y, n$ )

$c[i, j]$  = número de ocorrências de  $X[1..i]$  em  $Y[1..j]$

$$c[i, j] = \begin{cases} c[i-1, j] + c[i, j-1] & \text{se } X[i] = Y[j] \\ c[i, j-1] & \text{se } X[i] \neq Y[j] \end{cases}$$

$$c[0, j] = 1$$

SUBSEQ-CONT( $X, m, Y, n$ )

Para  $i = 0$  até  $m$

$$c[i, 0] \leftarrow 0$$

$$c[0, i] \leftarrow 1$$

Para  $i = 1$  até  $m$

Para  $j = 1$  até  $n$

se  $X[i] = Y[j]$

$$c[i, j] \leftarrow c[i-1, j] + c[i, j-1]$$

$$\text{senão } c[i, j] \leftarrow c[i, j-1]$$

devolver  $c[m, n]$

|   | D | A | B | B | B | I | T |
|---|---|---|---|---|---|---|---|
| R | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| A | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| B | 0 | 0 | 1 | 1 | 2 | 3 | 3 |
| B | 0 | 0 | 0 | 0 | 1 | 3 | 3 |
| I | 0 | 0 | 0 | 0 | 0 | 3 | 3 |
| T | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

Subestrutura ótima:

Lemma: Se  $Z_m$  é uma solução ótima para a instância  $X[1..m]$  e  $Y[1..n]$

1- Se  $X[m] = Y[n]$  então  $Z_{m-1}$  é solução com  $k-1$  subsequências de  $X[1..m-1]$  e  $Y[1..n]$

2- Se  $X[m] \neq Y[n]$  então  $Z_{m-1}$  é solução com  $k$  subsequências de  $X[1..m]$  e  $Y[1..n-1]$

1- Suponha que  $Z_{m-1}$  não é solução de  $X[1..m-1]$  e  $Y[1..n]$ .  
 Então existe outra solução ótima para  $X[1..m-1]$  e  $Y[1..n]$  tal que o nº de subsequências é maior que  $k-1$  e uma forma se adicionarmos o elemento  $n$  a solução terá mais que  $k$  subsequências, contradizendo.  
 2- Suponha que  $Z_{m-1}$  não é solução de  $X[1..m]$  e  $Y[1..n-1]$ .  
 Então, existe uma outra solução ótima para  $X[1..m]$  e  $Y[1..n-1]$  tal que o nº de subsequências é maior que  $k$ , uma contradição.

PROVA 2004/1

Questão - máximo - Recorrência

a) Recorrência comum

$$T(n) = T(n-1) + 1$$

$$T(1) = 1$$

Recorrência Média

$$T(1) = 1$$

$$T(n) = T(n-1) + E[X] \rightarrow \text{acho mesmo executado}$$

$$E[X] = X_1 + X_2 + X_3 + \dots$$

$X_i$  = a probabilidade de  $i$  ser o máximo  
 $= \frac{1}{i}$

$$E[X] = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots$$

$$= \sum_{i=1}^n \frac{1}{i} = \log(n)$$

$$X_n = \frac{1}{n}$$

$$X_{n-1} = \frac{1}{n-1}$$

$$X_1 = \frac{1}{1}$$

$$T(1) = 1$$

$$T(n) = T(n-1) + 1/n$$



$$b) T(n) = 1 \quad n=1$$

$$T(n) = T(n-1) + \frac{1}{n} \quad n \geq 1$$

$$T(n) = T(n-1) + \frac{1}{n}$$

$$= T(n-2) + \frac{1}{n-1} + \frac{1}{n}$$

$$= T(n-3) + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n}$$

$$= \dots$$

$$= T(1) + \sum_{k=2}^n \frac{1}{k}$$

$$= 1 + \sum_{k=2}^n \frac{1}{k}$$

$$= 1 + \sum_{k=1}^n \frac{1}{k} - 1$$

$$= \lg n + 1$$

Prova por indução em  $n$

base:  $n=1$

$$T(1) = 1 = \lg 1 + 1 = 1$$

passo:  $n > 1$

$$T(n) = T(n-1) + \frac{1}{n}$$

$$\leq \lg(n-1) + 1 + \frac{1}{n}$$

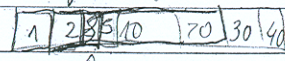
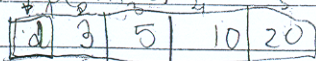
$$= \lg n - \lg\left(\frac{n}{n-1}\right) + 1 + \frac{1}{n}$$

$$= \lg n + 1 - \lg\left(\frac{n}{n-1}\right) + \frac{1}{n}$$

$$\leq \lg n + 1$$

## Questão 2 - LISTAS ORDENADAS

Tempo:  $O(m+n)$



PARA  $(A, m, B, n)$

- 1  $i \leftarrow 1$   $j \leftarrow 0$  se  $m=0$  devolva 1 se  $n=0$  devolva 0
- 2 enquanto  $j \leq n$  e  $B[j] \neq A[i]$
- 3  $j \leftarrow j+1$
- 4 se  $j > n$
- 5 devolva 0
- 6  $j \leftarrow j+1$   $i \leftarrow i+1$
- 7 enquanto  $j \leq n$  e  $i \leq m$
- 8 se  $B[j] = A[i]$
- 9  $i \leftarrow i+1$
- 10  $j \leftarrow j+1$
- 11 se  $i-1 = m$
- 12 devolva 1
- 13 devolva 0

Consumo

linha

2-3  $O(n)$

4-6  $O(1)$

7-10  $O(n)$

11-13  $O(1)$

consumo total de  $O(n)$

Corretude

As linhas 2-3 buscam o primeiro elemento de  $A$  em  $B$ .  
As linhas 4-10 buscam cada elemento de  $A$  em  $B$ , considerando que as pesquisas são fixas e as listas crescentes. Se todos os elementos de  $A$  foram encontrados em  $B$ , então  $A$  está em  $B$ .

Início:  $i-1 = 1$  então o vetor  $A[1..1]$  está em  $B$ .

Mantém:  $i-1 = k$  caso  $A[k]$  esteja em  $B$ ,  $i$  é adicionado.



1 1  
 1.  $j$  sempre adiciona o próximo elemento de  $B$ , mantendo a condição de que  $0 \leq A[i-1]$  está numericamente em  $B$ .

Termino:  $i-1 = n$  então o vetor  $A[1..n]$  está totalmente em  $B$ . O que conclui a prova do invariante.

Questão 3 - Para que valores reais da constante  $\alpha$  é verdade que  $n^\alpha$  cresce mais que  $\lg n$ ? Em outras palavras, para que valores reais de  $\alpha$  vale:  $\lg n = o(n^\alpha)$ ?

$$\lg n < \frac{1}{2} n^\alpha \text{ para } n \geq 1$$

$$\sqrt{\lg n} < n$$

$$\lg n^{\frac{1}{2}} < n$$

$$\frac{1}{2} \lg n < n$$

$$\lg n < \alpha n$$

$$\alpha > \frac{\lg n}{n} \text{ para } n \geq 1$$

$$\alpha > \frac{\lg 1}{1}$$

$$\alpha > 0$$

Questão 4 -  $P=NP$  ou não se algum problema em  $P$  é NP-completo. Essa afirmativa é verdadeira ou falsa? Justifique.

Verdadeira.  
 Se  $P=NP$  significa que todos os problemas em NP poderiam ser resolvidos em tempo polinomial, então todos os problemas NP poderiam ser reduzidos polinomialmente a  $P$ , ou seja, seriam NP-completos.

1 1  
 E se  $P$  está em NP-completo, significa que qualquer problema em NP pode ser reduzido polinomialmente a um problema  $P$ , tendo portanto uma solução em tempo polinomial, não havendo portanto diferença entre  $P$  e NP sendo  $P=NP$ .

Questão 5 - Descreva um algoritmo de programação dinâmica que dados um conjunto  $A$  com  $n$  inteiros positivos e um inteiro  $S$ , determina em tempo  $O(nS)$  se existe algum subconjunto de  $A$  cuja soma é  $S$ .

$$e[i,j] = \begin{cases} 1, & j=0 \\ 0, & i=0 \text{ e } j>0 \\ \max\{e[i-1,j], A[i] + e[i-1, j-A[i]]\} \end{cases}$$

|   | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 1 | 1 | 0 | 0 |

SOMA(A, n, S)

1 Para  $i \leftarrow 0$  até  $n$

2  $e[i, 0] \leftarrow 1$

3 Para  $j \leftarrow 1$  até  $S$

4  $e[0, j] \leftarrow 0$

5 Para  $j \leftarrow 1$  até  $S$

6 Para  $i \leftarrow 1$  até  $n$

7 Se  $A[i] > j$

8 então  $e[i, j] \leftarrow e[i-1, j]$

9 senão se  $e[i-1, j] > e[i-1, j-A[i]]$

10  $e[i, j] \leftarrow e[i-1, j]$

11  $e[i, j] \leftarrow e[i-1, j-A[i]]$

12

13 devolva  $e[n, S]$



Tamanho da instância  $(n, S)$

Consumo  $\gamma(n, S) = O(n \cdot S)$

As loops nas linhas 5-6 formam um consumo das linhas 4-11  $O(n \cdot S)$ .

Corretude

Subestrutura ótima

Suponha que  $K$  é solução ótima da instância  $(n, S)$

1- se  $n \in K$  então  $K - \{n\}$  é solução da instância  $(n-1, S - A[n])$

2- se  $n \notin K$  então  $K$  é solução da instância  $(n-1, S)$

Prova

1- Suponha que  $K - \{n\}$  não é solução ótima de  $(n-1, S - A[n])$  então existe uma outra solução ótima que a substitui e adicionando a esta nova solução ótima o elemento  $A[n]$  temos uma nova solução ótima melhor que  $K$  uma contradição.

2- Suponha que  $K$  não é solução ótima da instância  $(n-1, S)$  então eu posso concluir que  $K$  não é solução ótima de  $(n, S)$  o que é uma contradição.

Questão 6- MINIMIZAR  $d(x, y, z)$

a) MINIMIZAR  $(A, n)$   $\min \infty$

Para  $i \leftarrow 1$  até  $n$

Para  $j \leftarrow 1$  até  $n$

Para  $k \leftarrow 1$  até  $n$

$\text{se } A[i] \neq A[j] \neq A[k]$

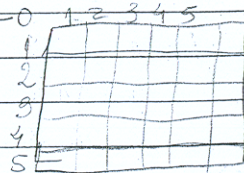
$d \leftarrow A[i] - A[j] + A[j] - A[k] + A[k] - A[i]$

$\text{se } \min > d$

então  $\min \leftarrow d$

$x \leftarrow A[i], y \leftarrow A[j], z \leftarrow A[k]$

devolva  $(x, y, z)$



b) MINIMIZAR  $(A, n)$

1  $A \leftarrow \text{HEAPSORT}(A, 1, n)$   $d \leftarrow \infty$   $x, y, z \leftarrow 0$

2  $i \leftarrow 1$   $j \leftarrow 3$

3 enquanto  $i \leq n-4$  e  $j < n+1$

4  $\text{se } A[i] \neq A[i+1] \neq A[j]$

5  $d \leftarrow A[i] - A[i+1] + A[i+1] - A[j] + A[j] - A[i]$

6  $\text{se } \min > d$

7 então  $\min \leftarrow d$

8  $x \leftarrow A[i]$

9  $y \leftarrow A[i+1]$

10  $z \leftarrow A[j]$

11  $i \leftarrow i+1$

12  $j \leftarrow j+1$

13 devolva  $(x, y, z)$

Consumo tamanho da instância  $= (n)$   
 linha consumo total  $= O(n \lg n)$   
 portanto  $= O(n^3)$

1  $O(n \lg n)$   
 2  $O(1)$   
 3-12  $O(n)$   
 13  $O(1)$

Corretude

Escolha gulosa

Se  $K$  é uma solução ótima então  $K < y < z \in K$ .

Prova

Suponha que  $K^*$  é uma solução ótima mas  $K < y < z$

se  $K > y$  e  $K > z$  e  $y > z$

isso quer dizer que  $z < y < K$

então podemos representar  $K \leftrightarrow z$  então teremos  $K < y < z$

se  $K > y$  e  $y < z$  e  $z > K$ , temos  $y < K < z$  e trocamos



$x < y$  temos  $x < y < z$   
 se  $x < y$ ,  $y > z$  e  $z > x$  então temos  $x < z < y$   
 mas se trocarmos  $y \leftrightarrow z$  temos  $x < y < z$

### Subestrutura ótima

- Lembrar: Se  $K$  uma solução ótima para  $A[1..n]$  com

$x < y < z \in A$

1- Se  $z = K_n$  então  $K_{n-1}$  é solução ótima para  $A[1..n-1]$  com  $x < y < z \in A[1..n-1]$

2- Se  $z \neq K_n$  então  $K$  é solução ótima para  $A[1..n-1]$  com  $x < y < z \in A[1..n-1]$

### Prova

1- Suponha que  $K_{n-1}$  não é solução ótima para  $A[1..n-1]$  com  $x < y < z \in A$ , então existe uma outra solução para  $A[1..n-1]$  que adicionando o elemento  $n$  forma uma outra solução ótima para  $A[1..n]$  diferente de  $K$  o que é uma contradição

2- Suponha que  $K^*$  não é solução ótima para  $A[1..n-1]$  com  $x < y < z \in A$ , concluímos de forma direta que  $K^*$  não é solução ótima para  $A[1..n]$ , uma contradição.

### Questão 4 - ÁRVORE

EMBTIDA(A, n, B, m)

1 se  $n=0$

2 retorna 1

3  $i \leftarrow \text{BUSCA}(B, A[0])$  se  $i = -1$  retorna 0

4 se  $\text{EH-EMBTIDA}(A, B, 1, i) > 0$

5 retorna 0

6 retorna 1

$$2T(n-1) + 1$$

EH-EMBTIDA(A, B, ra, rb)

se  $ra \neq \text{NIL}$  e  $rb \neq \text{NIL}$  retorna 1  
 se  $ra \neq \text{NIL}$  e  $rb = \text{NIL}$  retorna 0  
 se  $ra = \text{NIL}$  e  $rb \neq \text{NIL}$  retorna 0

$c \leftarrow \text{EH-EMBTIDA}(A, B, \text{dir}[ra], \text{dir}[rb])$

se  $c > 0$

retorna c

se  $A[ra] \neq B[ra]$

retorna  $c + 1$

$c \leftarrow \text{EH-EMBTIDA}(A, B, \text{dir}[ra], \text{dir}[rb])$

se  $c > 0$

retorna c

retorna 0

Lembrando: Tamanho da instância:  $(n, m)$

EMBTIDA

linha tamanho

1-2  $O(1)$

3  $O(\lg m)$

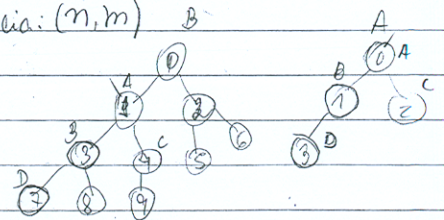
4  $O(n)$

5-6  $O(1)$

Total:  $O(\lg m + n)$

Considerando a altura da

$B = \lg m$



EH-EMBTIDA

Tamanho da instância: n° de nós de A

Notação:  $n$

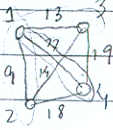
$T(n) = T(k) + T(n-k-1) + \Theta(1)$

$0 \leq k \leq n-1$

$T(n) = \Theta(n)$



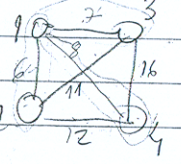
# Questão 8 - Kruskal Kruskal

1-   $1+4 \times 2$   $5+6 \times 6$   $n^{\circ} \text{ vertices} = 6$   $n^{\circ} \text{ aresta} = \frac{n^2 - n}{2}$

1-  $O(n)$   $2 - O(E)$  build-HEAP  $3 - 4 - 5 - O(E)$  Procura aresta menor

$V = n$  e  $E = \frac{n^2 - n}{2}$

1-2 Como não caso não recursivamente  
 2-3 As primeiras arestas menores são realmente  
 3-4  
 1-5 escolhidas então para não necessário procurar todas  
 1-6 as arestas.  $O(\frac{n^2 - n}{2} + \frac{n^2 - n}{2}) = \frac{n^2 - n}{2} = O(n^2)$

2-   $3+4 \times 1$   $2+4 \times 1$   $4+4 \times 1$   $4+4 \times 2$   $3+4 \times 2$   $4+4 \times 3$

$A = \frac{4^2 - 4}{2} = \frac{16 - 4}{2} = \frac{12}{2} = 6$

1-  $O(1)$   $2 - O(V \cdot \lg E)$   $3 - 5 - O(E)$

1-2 Sendo  $V = n$   
 1-5  $E = \frac{n^2 - n}{2}$   
 1-6

2-3 Como não caso  
 3-4 As 1<sup>as</sup> arestas são escolhidas então  
 no máximo vão ser procuradas  $n$  arestas  
 $O(\frac{n^2 - n}{2} + n) = \frac{n^2 - n + 2n}{2} = \frac{n^2 + n}{2} = O(n^2)$

NOME: \_\_\_\_\_  
 E-MAIL: \_\_\_\_\_  
 MATÉRIA: \_\_\_\_\_  
 PROFESSOR: \_\_\_\_\_ SALA: \_\_\_\_\_

| HORÁRIOS |         |       |        |        |       |        |
|----------|---------|-------|--------|--------|-------|--------|
| HORÁRIO  | SEGUNDA | TERÇA | QUARTA | QUINTA | SEXTA | SÁBADO |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |
|          |         |       |        |        |       |        |

| PROVAS/NOTAS |  | TRABALHOS |  |
|--------------|--|-----------|--|
|              |  |           |  |
|              |  |           |  |
|              |  |           |  |
|              |  |           |  |
|              |  |           |  |
|              |  |           |  |
|              |  |           |  |

NOTAS: \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_