Leonardo Rodrigues Marques RA: 178610

Leonardo Rodrigues Marques

Corrigir questão 01.

1) 4,0

① <u>Base</u>: Considere dois vetores X e Y de tamanho $n=2$. Ordene $X \cup Y$ em um vetor V de tamanho $n=4$. O elemento mediano estará nas posições $\frac{n}{2}$ ou $\frac{n}{2}+1$. do vetor V de tamanho $n=4$.

<u>P.H.I</u>: Dado um vetor V genérico com $k<n$ elementos, sabemos calcular o valor mediano desse vetor V.

<u>Passo</u>: Considere os vetores X e Y de tamanho $n>2$. Na busca pela mediana de $X \cup Y$, temos dois casos:

- <u>n ímpar</u>: os elementos medianos m_x e m_y de X e Y são exatamente os elementos centrais $\lceil \frac{n}{2} \rceil$ dos vetores X e Y, respectivamente. Usando o paradigma da divisão e conquista, temos sub-casos:
  - $m_x > m_y$: a mediana de $X \cup Y$ está inserida ou do primeiro (x_i) elemento de X até m_x ou de m_y até o último elemento de Y (y_f).

Por quê?

  - $m_x < m_y$: a mediana de $X \cup Y$ está inserida ou de m_x até o último elemento de X (x_f) ou do primeiro elemento de Y (y_i) até m_y.

Por quê?

Para ambos os ~~instâncias~~ casos, temos instâncias $\frac{n}{2}$ menores e sabemos calcular suas medianas.

Por que a mediana da união dos subarrays é mediana da união de X e Y?

com $m_{x<} < m_{x>}$.

- <u>n par</u>: os elementos medianos de X são $m_{x<}$ e $m_{x>}$ e de Y são $m_{y<}, m_{y>}$ com $m_{y<} < m_{y>}$ e estão nas posições centrais dos vetores. Usando o paradigma de divisão e conquista, temos os seguintes casos:

- $m_{X>} < m_{Y<}$: a mediana de $X \cup Y$ está inserida ou de $m_{X<}$ até o último elemento de X (x_f) ou do primeiro elemento de Y (y_i) ~~até~~ até $m_{Y<}$.

- $m_{X>} > m_{Y<}$: a mediana de $X \cup Y$ está inserida ou do primeiro elemento de X (x_i) até $m_{X<}$ ou de $m_{Y<}$ até o último elemento de Y (y_f).

Para ambos os casos, obtemos instâncias $\frac{n}{2}$, e sabemos calcular suas medianas ■.

OBS:

$X: \; 1 \;\; 2 \;\; \boxed{3} \;\; 4$ ou $X: \; 6 \;\; \boxed{7} \;\; 8 \;\; 9$

$Y: \; 6 \;\; \boxed{7} \;\; 8 \;\; 9$ $\qquad Y: \; 1 \;\; 2 \;\; \boxed{3} \;\; 4$

Como usamos o paradigma de divisão e conquista, temos complexidade $O(\lg n)$. Na base, tem-se uma ordenação de um vetor de tamanho fixo e pequeno, como é constante, podemos desprezar seu tempo. Por definição, $f(n) \leq O(\lg n) < o(n)$, temos um algoritmo limitado no pior caso por o-zinho (n).

(2). O problema de determinar um elemento mediano $X \cup Y$ com X e Y vetores ordenados é basicamente um problema de busca em um vetor ordenado baseado em comparações e ele pode ser representado através de árvores de decisão. 2) Não corrigido.

- Cada nó representa uma comparação entre medianas.
- As ramificações correspondem aos resultados comparados.
- As folhas correspondem a possíveis respostas do algoritmo.
- A árvore de altura h deve ter pelo menos $2n$ folhas e tem no máximo 2^h folhas, portanto $2^h \geq 2n$, $h \geq \log_2 2n$.
- Logo, a altura da árvore é pelo menos $\Omega(\log_2 2n)$, o que permite concluir que a busca ~~do~~ em vetor ordenado tem cota inferior $\Omega(\log_2 2n)$.

③ 3a) 1,0

ⓐ A complexidade do HeapSort é $O(n \lg n)$.

Xitória está errado pelo seguinte fato.

A soma das complexidades do Heap Sort nos k vetores é $\sum_{i=1}^{k} n_i \lg n_i$

é maior que $\sum_{i=1}^{k} n_i$ e $O(n)$.

$$\sum_{i=1}^{k} n_i \lg n_i > \sum_{i=1}^{k} n_i \; ① \Rightarrow \sum_{i=1}^{k} (n_i \lg n_i - n_i) > 0 \Rightarrow \sum_{i=1}^{k} n_i (\log n_i - 1) > 0 \; ②$$

Para satisfazer ②, $(\log n_i - 1) > 0$, portanto $\log n_i > 1$. Voltando em ①, como $\log_2 n_i > 1$, a parte esquerda é maior que a parte direita, o que prova que HeapSort não executa em $O(n)$ no pior caso.

ⓑ 3b) 1,0

A complexidade do Couting Sort é $O(n+k)$

Chorozinho está errado pelo seguinte fato.

Para um dado vetor a ser ordenado, é necessário um vetor auxiliar de tamanho n que consome tempo de pior caso $O(n)$. Como o problema, existem k vetores de tamanho n no pior caso, o que consome $n \, O(n)$, que é equivalente $O(n^2) > O(n)$.

$$O(n) \Rightarrow \sum_{i=1}^{k} (\underbrace{n_i}_{\text{k vetores}} + \underbrace{n}_{\text{vetor auxiliar}}) \Rightarrow \sum_{i=1}^{k} n_i + \sum_{i=1}^{k} n \Rightarrow O(n) + n \underbrace{\sum_{i=1}^{k} O(1)}_{\text{definição}}$$

$$\Rightarrow O(n) + n(O(n) \Rightarrow O(n^2).$$

3c) 0,0. Em branco.