

MO417 – Ata do Problema 9-1

Rafael Seraphin Seste

12 de abril de 2010

Enunciado: Dado um conjunto de n números, queremos encontrar os i maiores em sequência ordenada, usando um algoritmo baseado em comparação. Descubra o algoritmo que implementa cada um dos métodos a seguir com o melhor tempo de execução assintótico no pior caso e analise os tempos de execução dos algoritmos em termos de n e i .

- A. Classifique os números e liste os i maiores.
- B. Construa uma fila de prioridade a partir dos números e chame EXTRACT-MAX i vezes.
- C. Use um algoritmo de estatística de ordem para localizar o i -ésimo maior número, particionar e ordenar os i maiores números.

Resolução:

- A. Para ordenar utilizamos o MERGE-SORT no conjunto, resultando em um tempo de execução $O(n \cdot \lg n)$.
A listagem dos i maiores levará tempo i .
Como i sempre será menor que $n \cdot \lg n$ temos que:

$$T(n) = O(n \cdot \lg n).$$

- B. Constrói-se o heap, que executa em $O(n)$. Após isto, faz-se i vezes a extração do máximo, como cada execução do EXTRACT-MAX demora $O(\lg n)$, temos que as i extrações demoram $O(i \cdot \lg n)$.
Como não é possível determinar se $i \cdot \lg n$ é maior ou menor que n , o tempo de execução será:

$$T(n) = O(n + i \cdot \lg n).$$

- C. Utilizando os algoritmos na sequência abaixo:
 - SELECT (com tempo $O(n)$)
 - PARTITION (com tempo $O(n)$)
 - MERGE-SORT nos i maiores (com tempo $O(i \cdot \lg i)$)Como no item acima, não é possível determinar se $i \cdot \lg i$ é maior ou menor que n , então o tempo de execução será:

$$T(n) = O(n + i \cdot \lg i).$$