

Ata da resolução do problema 9.1-1

Washington Luís Pereira Barbosa
RA: 971766

16 de maio de 2009

1 Enunciado do problema

Mostre que o segundo menor entre n elementos pode ser encontrado com $n + \lg n - 2$ comparações no pior caso. (Sugestão: Encontre também o menor elemento.)

2 Idéia da resolução apresentada por Robson Peixoto

- Comparar dois a dois todos os elementos do vetor e guardar, em um novo vetor, os menores elementos dessas comparações. Assim teremos um novo vetor com $n/2$ elementos para comparar. Repetindo o processo anterior chegaremos a um único elemento, que será o menor elemento de todos. No final desses passos teremos feito $n - 1$ comparações;
- Se pensarmos nas comparações anteriores em forma de uma árvore binária teremos uma árvore com altura $\lg n$, e para encontrarmos o segundo menor número devemos fazer a comparação com $\lg n - 1$ elementos voltando na árvore binária com os elementos que “perderam” na comparação anterior;
- Assim teremos $(n - 1) + (\lg n - 1) = n + \lg n - 2$ comparações.

3 Algoritmo baseado na sugestão de Paulo Gurgel

Idéia do algoritmo:

- Foram feitas pequenas alterações e alguns detalhamentos do código original para melhor compreensão do mesmo;
- Tem-se um vetor de tamanho $2n - 1$;
- As n primeiras posições estão ocupadas com os n valores dos quais se deseja buscar os dois menores, e as $n - 1$ últimas posições estão livres. Cada posição do vetor possui o valor e um ponteiro para o último valor com o qual foi comparado e que foi o menor na comparação. Inicialmente estes valores do ponteiro são NIL.
- O segredo deste algoritmo é fazer com que estes ponteiros formem uma lista encadeada de todos os valores menores em comparações anteriores.

Então, para o menor elemento, funciona assim:

- Os valores dos elementos são comparados dois a dois (linha 2);
- Pega-se o elemento de menor valor e coloca na primeira posição livre do vetor, $[n + i]$ (linha 3 ou 7).;

- Adiciona na lista encadeada deste elemento o valor do elemento com o qual ele foi comparado (linhas (4 e 5) ou (8 e 9));
- Quando o laço acabar, o menor valor estará na posição $2n - 1$.

Para o segundo menor elemento, utilizaremos a lista encadeada do último elemento, pois esta contém todos os valores que foram comparados com ele

- O While da linha 16 percorre a lista encadeada do último elemento atrás do valor do menor elemento com o qual ele foi comparado;
- Quando o laço acabar, o segundo menor valor do vetor estará na variável.

Algoritmo 1: Algoritmo para o exercício 9.1-1

```

/* Achar e apresentar o menor valor enquanto monta a lista encadeada para a
   busca do segundo menor */
1 for i ← 1 to n - 1 do
2   if A[2i - 1].valor < A[2i].valor then
3     A[n + i].valor ← A[2i - 1].valor
4     A[2i].ponteiro ← A[2i - 1].ponteiro
5     A[n + 1].ponteiro ← 2i
6   else
7     A[n + i].valor ← A[2i].valor
8     A[2i - 1].ponteiro ← A[2i].ponteiro
9     A[n + 1].ponteiro ← 2i - 1
10  end
11 end
12 menor ← A[2n - 1].valor
13 print(menor)

/* Buscar e apresentar o segundo menor valor a partir da lista encadeada */
14 proximo ← A[2n - 1].ponteiro
15 menor2 ← A[proximo].valor
16 while A[proximo].ponteiro ≠ NIL do
17   proximo ← A[proximo].ponteiro
18   if menor2 > A[proximo].valor then
19     menor2 ← A[proximo].valor
20   end
21 end
22 print(menor2)

```
