

Atividade Métodos de Ordenação

Mauricio Rodrigues Lima

21 de fevereiro de 2024

1 Otimize o Algoritmo

Seja o algoritmo a seguir:

```
void bubble_sort(unsigned long long int *vetor){  
  
    unsigned long long int i, j, aux;  
  
    for(i = 0; i < TAMANHO_VETOR; i++){  
        for(j = 0; j < TAMANHO_VETOR; j++){  
            if(vetor[i] < vetor[j]){  
                aux = vetor[i];  
                vetor[i] = vetor[j];  
                vetor[j] = aux;  
            }  
        }  
    }  
}
```

Otimize esse algoritmo para que possua o mínimo de comparações possível. Demonstre sua otimização.

2 Pontuação de Competidores

Em uma competição, cada competidor recebe uma pontuação com base no desempenho em diferentes eventos. Sua tarefa é classificar os competidores de acordo com suas pontuações totais. Se dois ou mais competidores tiverem a mesma pontuação total, eles devem ser classificados com base em seus IDs de competidor em ordem decrescente.

2.1 Entrada

- A primeira linha contém um inteiro N ($1 < N < 1000$), representando o número de competidores.
- As próximas N linhas contêm as informações de cada competidor no formato "ID PontuaçãoEvento1 PontuaçãoEvento2 ... PontuaçãoEventoM", onde ID é um número inteiro único que identifica o competidor, e as Pontuações dos Eventos são números inteiros representando a pontuação obtida pelo competidor em cada evento. Cada competidor tem pelo menos 1 e no máximo 10 eventos.

2.2 Saída

- N linhas contendo os IDs dos competidores, ordenados de acordo com as regras especificadas.

2.3 Exemplo

Entrada:

```
4
101 20 15 30
102 25 30 20
103 30 25 25
104 20 25 20
```

Saída:

```
104
101
102
103
```

3 Erros de Algoritmos

Quais os erros dos algoritmos de ordenação a seguir? Apresente os erros e como corrigi-los.

a)

```
void selectionSort(int arr[], int n) {
    int i, j, min_idx;
    for (i = 0; i <= n-2; i++) {
        min_idx = 0;
        for (j = i+1; j < n; j++) {
            if (arr[j] < arr[min_idx]) {
                min_idx = j
            }
        }
        int temp = arr[i];
        arr[min_idx] = arr[0];
        arr[min_idx] = temp;
    }
}
```

b)

```
void insertionSort(int arr[], int n) {
    int i, key, j;
    for (i = 1; i <= n-1; i++) {
        key = arr[i];
        j = i - 1;
        while (j > 0 && arr[j] >= key) {
            arr[j] = arr[j + 1];
            j = j + 1;
        }
        arr[j + 1] = key;
    }
}
```

4 Implementação de Métodos de Ordenação

Demonstre o tempo de execução de cada algoritmo de ordenação apresentado em sala, considerando grandes vetores ordenados de forma decrescente e que devem ser ordenados de forma crescente. Utilize a função `clock()` já apresentada em sala.

“Não são nossas habilidades que mostram quem somos, mas nossas escolhas.” - Dumbledore