16/14

Segunda Prova de Algoritmos e Estruturas de Dados II
CI056-D
27/01/2017

1) Considere o códido em C abaixo, que implementa o algoritmo de ordenação Quicksort. A função "quicksort" recebe como parâmetros o vetor "v" e as posições de início "a" e fim "b" da região do vetor a ser ordenada. A função "particiona", usa o último elemento do vetor "v[b]" como pivo para particionar o vetor "v" e retorna o índice da posição do pivo em "v".

```
void quicksort(int v[], int a, int b) {
    if(a < b) {
        p = particiona(v, a, b);
        quicksort(v, a, p-1);
        quicksort(v, p+1, b);
    }
}

int particiona(int v[], int a, int b) {
    int pivo, i, j;

    pivo = v[b];
    i = a;
    for(j = a; j < b; j++) {
        if(v[j] <= pivo) {
          if(i != j)
                troca(v, i, j);
          i++;
      }
    }
    troca(v, i, b);
    return i;
}
```

Explique em que casos este algoritmo executa um número de operações proporcional a $n^2$. Apresente um vetor com pelo menos 5 elementos que seja um exemplo destes casos.

2) Escreva uma função em C que inverte os elementos de uma fila usando uma pilha. A função deve usar apenas as operações básicas de fila e pilha: enfilera, desenfilera, fila_vazia, empilha, desempilha e pilha_vazia.

parâmetros, retorno de funções, etc.

3) A sequência dos "números de granizo" a partir de um determinado número "x" é dada pelas seguintes regras:

- se "x" é igual a 1, a sequência termina;
- se "x" é par, o próximo número da sequência é "x / 2";
- se "x" é impar, o próximo número da sequência é "3 * x + 1"

Escreva uma função recursiva em C que recebe um inteiro "x" e imprime a sequência dos "números de granizo" a partir de "x".