UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá – Ciência da Computação
Prof. Atílio Luiz
QXD0010- Estrutura de Dados

UFC

AP1
2025.2

Nome: ________________________________ Matrícula: ______________

[2.5 pontos] 1. Logo abaixo apresentamos um algoritmo de ordenação chamado `ExchangeSort`. Sobre esse algoritmo, responda as questões abaixo:

(a) Descreva quem é a entrada de pior caso desse algoritmo e justifique;

(b) Calcule formalmente a quantidade de passos que esse algoritmo realiza no pior caso;

(c) Quais as diferenças entre esse algoritmo e o BubbleSort e o SelectionSort?

(d) Esse algoritmo é estável? Justifique.

```
1  void exchangeSort(int n, int A[]) {
2      for (int i = 0; i < n-1; i++) {
3          for (int j = i + 1; j < n; j++) {
4              if (A[j] < A[i]) {
5                  int aux = A[i];
6                  A[i] = A[j];
7                  A[j] = aux;
8              }
9          }
10     }
11 }
```

Solução:

(a) A entrada de pior caso está em ordem decrescente. Justificativa: essa entrada fará com que o teste da linha 4 resulte sempre verdadeiro, o que causará uma troca de valores.

(b) A quantidade de passos realizada pela linha 4 dá a complexidade de pior caso do algoritmo, calculada abaixo:

$$\begin{aligned} T(n) &= \sum_{i=0}^{n-2}\left(\sum_{j=i+1}^{n-1} c\right) = \sum_{i=0}^{n-2} c(n-i-1) = \sum_{i=0}^{n-2} cn - \sum_{i=0}^{n-2} ci - \sum_{i=0}^{n-2} c \\ &= cn(n-1) - \frac{c(n-2)(n-1)}{2} - c(n-1) \\ &= \frac{1}{2}(cn^2 - cn) \end{aligned}$$

(c) O Bubblesort visto em sala trata de empurrar os menores elementos para o início vindo de trás para frente trocando elementos vizinhos, já o Exchangesort empurra os menores para o início indo do início para trás, usando o elemento A[i] como referência nas comparações. O Selectionsort faz apenas uma troca ao final, já o Exchangesort faz troca toda vez que ele encontra um menor elemento em posição invertida.

(d) Não é estável. Por exemplo, execute o vetor $A = [4, 4, 3]$.

[2.5 pontos] 2. Em sala de aula, vimos diversos algoritmos de ordenação. Dentre eles, vimos o algoritmo `MergeSort` que é um algoritmo de complexidade $O(n \lg n)$. Esse algoritmo usa como procedimento auxiliar o algoritmo de intercalação. Escreva em linguagem C uma **versão recursiva** do algoritmo da intercalação. Antes, convém reformular o problema da seguinte maneira: dados como entrada dois vetores crescentes $U[0..m-1]$ e $V[0..n-1]$, produzir como saída um vetor crescente $W[0..m+n-1]$ que contenha o resultado da intercalação dos dois vetores.

Atenção: Escreva um comentário explicando cada parâmetro de entrada da função. Deixe claro quem é a saída da função. Forneça a assinatura completa da função (nome da função, tipo de retorno, os tipos dos parâmetros de entrada). Ademais, escreva a função usando a sintaxe correta do C.

Solução: A função se chama **IntercalaRecursivo** e possui 8 parâmetros de entrada:

- os arrays U, V e W
- m: o tamanho do array U
- n: o tamanho do array V
- i: em que posição do array U está a leitura
- j: em que posição do array V está a leitura
- k: em que posição do array W está a leitura

Como a função já recebe como entrada o vetor W ela não retorna nada.

```c
void IntercalaRecursivo(int U[], int m, int i, int V[],
                        int n, int j, int W[], int k)
{
    if(i == m && j == n)
        return;

    if(i == m && j < n) {
        W[k] = V[j];
        IntercalaRecursivo(U, m, i, V, n, j + 1, W, k + 1);
    }
    else if(i < m && j == n) {
        W[k] = U[i];
        IntercalaRecursivo(U, m, i + 1, V, n, j, W, k + 1);
    }
    else if(U[i] <= V[j]) {
        W[k] = U[i];
        IntercalaRecursivo(U, m, i + 1, V, n, j, W, k + 1);
    }
    else {
        W[k] = V[j];
        IntercalaRecursivo(U, m, i, V, n, j + 1, W, k + 1);
    }
}
```

Boa Prova!

[2.5 pontos] 3. Escreva em linguagem C uma função que rearranje um vetor $V[p..r]$ de números inteiros de modo que os elementos negativos e nulos fiquem à esquerda e os positivos fiquem à direita. Em outras palavras, rearranje o vetor de modo que tenhamos $V[p..j-1] \leq 0$ e $V[j..r] > 0$ para algum j tal que $p \leq j \leq r+1$. Escreva uma função eficiente que não use vetor auxiliar.

Atenção: Escreva um comentário explicando cada parâmetro de entrada da função. Deixe claro quem é a saída da função. Forneça a assinatura completa da função (nome da função, tipo de retorno, os tipos dos parâmetros de entrada). Ademais, escreva a função usando a sintaxe correta do C.

Solução:

```c
int rearranjo(int v[], int p, int r) {
    int j = p;
    for(int k = p; k <= r; k++) {
        if(v[k] <= 0) {
            int aux = v[j];
            v[j] = v[k];
            v[k] = aux;
            j++;
        }
    }
    return j;
}
```

Boa Prova!

[2.5 pontos] 4. Para cada uma das afirmações abaixo diga explicitamente se ela é Verdadeira ou Falsa. Mais importante que isso, se for Verdadeira então prove formalmente. Se for Falsa, então prove formalmente também. **Atenção:** Para essa questão está proibido usar limites. Deve-se aplicar a definição de notação-O vista em sala.

(i) $3n^5 + 10n^4 - 4n^3 + 17n^2 + 80 = O(n^5)$

Solução: Verdadeiro. Note que a desigualdade abaixo vale para todo $n \geq 1$:

$$3n^5 + 10n^4 - 4n^3 + 17n^2 + 80 \leq 3n^5 + 10n^5 + 4n^5 + 17n^5 + 80n^5 = 114n^5$$

Logo, $3n^5 + 10n^4 - 4n^3 + 17n^2 + 80 \leq cn^5$ para $c = 114$ e $n \geq n_0 = 1$.
Portanto, $3n^5 + 10n^4 - 4n^3 + 17n^2 + 80 = O(n^5)$. ■

(ii) $3n^2 - 5n + 60 = O(n^2)$

Solução: Verdadeiro. Note que a desigualdade abaixo vale para todo $n \geq 1$:

$$3n^2 - 5n + 60 \leq 3n^2 + 5n^2 + 60n^2 = 68n^2$$

Logo, $3n^2 - 5n + 60 \leq cn^2$ para $c = 68$ e $n \geq n_0 = 1$.
Portanto, $3n^2 - 5n + 60 = O(n^2)$. ■

(iii) $n^2 - 200n - 300 = O(n)$

Solução: Falso. Queremos provar que $n^2 - 200n - 300 \neq O(n)$. Vamos provar por contradição. Suponha, por absurdo, que $n^2 - 200n - 300 = O(n)$. Então, pela definição de notação O, existem constantes positivas c e n_0 tais que $n^2 - 200n - 300 \leq cn$ para todo $n \geq n_0$. Simplificando a desigualdade, obtemos:

$$\begin{aligned}
n^2 - 200n - 300 &\leq cn \\
\frac{n^2 - 200n - 300}{n} &\leq \frac{cn}{n} \\
n - 200 - \frac{300}{n} &\leq c \\
n - \frac{300}{n} &\leq c + 200
\end{aligned}$$

Note que todo valor de n maior ou igual a $c + 300$ torna esta última desigualdade falsa. Logo, não é verdade que vale para todo n no infinito. Contradição. Logo, nossa suposição inicial é falsa. O correto é que $n^2 - 200n - 300 \neq O(n)$. ■

(iv) $2n \lg n - 10n + 100 \lg n = O(n \lg n)$

Solução: Verdadeiro. Note que a desigualdade abaixo vale para todo $n \geq 2$:

$$2n \lg n - 10n + 100 \lg n \leq 2n \lg n + 10n \lg n + 100n \lg n = 112n \lg n$$

Logo, $2n \lg n - 10n + 100 \lg n \leq cn \lg n$ para $c = 112$ e $n \geq n_0 = 2$.
Portanto, $2n \lg n - 10n + 100 \lg n = O(n \lg n)$. ■

Boa Prova!