Universidade Federal de Santa Catarina - Campus Blumenau
Departamento de Engenharia de Controle, Automação e Computação **Avaliação 3**
BLU3202 — **Algoritmos e Estruturas de Dados** 2022/2

Aluno(a): ..........................................................................................................

1. Considere o vetor $v = [2, 4, 5, 10, 11, 14, 16, 17, 21, 22, 24, 27]$. Escreva a sequência de números avaliados na pesquisa binária pelos seguintes números:
11:____________________________________________
24:____________________________________________

2. Considere a o vetor $v = [4, 5, 7, 9, 11, 13, 15, 17, 19, 21, 24]$. Escreva a sequência de números avaliados na pesquisa binária pelos seguintes números:
10: ____________________________________________
23: ____________________________________________

3. Escreva o estado do vetor $v = [25, 18, 6, 26, 19, 24, 21, 2]$ após cada passo completo na execução do algoritmo de ordenação por inserção.

____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________

4. Escreva o estado do vetor $v = [21, 18, 19, 1, 17, 33, 27, 11]$ após cada um dos passos da execução do algoritmo de ordenação por seleção.

____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________

5. Considere o vetor $v = [53, 20, 19, 14, 39, 40, 42, 15, 31, 18]$. Ao final do processo de ordenação por *merge sort*, os dois vetores que, quando intercalados no processo de ordenação, resultarão no vetor $v$ ordenado são:

   (a) $[15, 14, 14, 20, 19]$ e $[31, 39, 40, 42, 53]$

   (b) $[14, 19, 20, 39, 53]$ e $[15, 18, 31, 40, 42]$

   (c) $[15, 19, 14, 20, 14]$ e $[39, 31, 40, 42, 53]$

   (d) $[14, 19, 53, 20, 39]$ e $[15, 40, 42, 14, 31]$

   (e) $[53, 39, 19, 20, 14]$ e $[42, 40, 31, 14, 15]$

6. Nas afirmativas a seguir a respeito da ordenação do vetor $v = [75, 17, 12, 88, 27, 71, 7, 60]$, assinale V para as verdadeiras e F para as falsas

   ( ) Na ordenação por *quick sort*, o número 7 encontrará sua posição definitiva antes do número 88. Considerar que o pivô é o elemento central do vetor.

   ( ) O número 7 levará menos passos para ocupar sua posição definitiva se o vetor $v$ for ordenado através da ordenação por seleção do que se for ordenado através da ordenação por inserção.

   ( ) Na ordenação por seleção, o número 17 encontra sua posição definitiva antes do número 7.

   ( ) Sendo $n$ o maior índice do vetor, no caso de ordenação por *merge sort*, o número 7 ocupará uma posição menor que $\lfloor \frac{n}{2} \rfloor$ apenas no vetor final, já ordenado.

7. Considere a aplicação do algoritmo *merge sort* ao vetor $v = [11, 28, 24, 29, 31, 23, 38, 26, 27, 21, 42, 37, 50, 40, 29, 10]$. Escreva abaixo os dois vetores que, quando intercalados no processo de ordenação, resultarão no vetor $v$ ordenado.

____________________________________________________
____________________________________________________

8. Aplique o algoritmo de ordenação *Quick sort* no vetor $v = [32, 18, 45, 25, 24, 49, 39, 27]$ utilizando como pivô $p$ o elemento central do vetor (ou seja, sendo $l$ e $r$ os índices das extremidades esquerda e direita do vetor respectivamente, considera-se $p = \lfloor \frac{l+r}{2} \rfloor$). Durante a ordenação, o subvetor mais à esquerda do pivô deve ser ordenado antes do subvetor mais à direita. Mostre cada um dos passos da ordenação que levaram a obter o vetor ordenado. Considere que um *passo de ordenação* está completo quando o pivô está em sua posição definitiva.

____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________

9. Assinale V para as alternativas verdadeiras ou F para as alternativas falsas com respeito ao algoritmo de ordenação *quick sort*:

   ( ) O pior caso, isto é, o caso em que o algoritmo tem o pior desempenho, é aquele em que os pivôs, ao final de um passo de ordenação, sempre ocuparão uma das extremidades do vetor.

   ( ) No melhor caso, o *quick sort* tem desempenho igual ao *merge sort*.

   ( ) Para que o quick sort apresente o melhor desempenho possível, o pivô deve ser sempre o elemento com o menor valor entre os elementos ainda não ordenados.

   ( ) Em um subvetor com n elementos, para $n \geq 3$, pode-se selecionar, como pivô, o elemento médio entre três valores quaisquer. Nesse caso, garante-se que o custo da ordenação será inferior ao pior caso, que é $O(n^2)$.

   ( ) A escolha como pivô do elemento que ocupa a posição central do vetor a ser ordenado garante que o algoritmo terá o melhor desempenho possível

   ( ) Se, em todas as iterações do algoritmo, o pivô for o elemento médio de um vetor com n elementos, então a ordenação terá custo de $O(n^2)$. Observação: por elemento médio, entende-se o elemento cujo valor seja o valor mediano entre todos os elementos do vetor. Por exemplo, o elemento médio do vetor v=[3,5,1,7,2,15,11] é 5.

   ( ) Considerando-se a análise assintótica, o desempenho do algoritmo quick sort depende do valor do pivô que foi escolhido independente da sua posição original (isto é, sua posição no vetor não ordenado).

   ( ) Mesmo no pior caso, o *quick sort* tem desempenho melhor que a ordenação por inserção e que a ordenação por seleção

10. Considere o código a seguir e complete as lacunas para que a função *mergeSort* implemente o agoritmo de ordenação MergeSort.

```
1  int *merge(int *v1, int t1, int *v2, int t2){
2     int *novo = malloc(sizeof(int)*(t1+t2));
3     int i1=0, i2=0, inovo=0;
4     while(i1<t1&i2<t2){
5        if(v1[i1]<v2[i2])
6           novo[inovo++] = v1[i1++];
7        else
8           novo[inovo++] = v2[i2++];
9     }
10    while(i1<t1) novo[inovo++]=v1[i1++];
11    while(i2<t2) novo[inovo++]=v2[i2++];
12    return novo;
13 }
14
15 void mergeSort(int *v, int inicio, int fim){
16    if(inicio<fim){
17       int meio = (inicio+fim)/2;
18       mergeSort(v,____, ____ );
19       mergeSort(v,____,fim);
20          int *novo=merge(v+inicio,meio-inicio+1,
21                          v+meio+1,fim-meio);
22          int i;
23          for(i=0;i<=fim-inicio;i++)
24             v[____] = novo[i];
25    }
26 }
```

Linha 18 (1): ______________________________________
Linha 18 (2): ______________________________________
Linha 19: ______________________________________
Linha 24: ______________________________________

### Informações úteis

- Em vetores com número par de elementos, considerar, como elemento central, o último elemento da primeira metade.
- Quando dois subvetores precisarem ser ordenados, considerar que o subvetor da esquerda é ordenado antes do subvetor da direita.
- Ao dividir um vetor $v = [v_0, \cdots, v_n]$ pela metade, sendo 0 (zero) o índice do primeiro elemento e $n$ o índice do último elemento, considerar que (i) a primeira metade é $[v_0, \cdots, v_c]$ e (ii) a segunda metade é $[v_{c+1}, \cdots, v_n]$, onde $c = \lfloor \frac{0+n}{2} \rfloor$.
- Em um algoritmo de ordenação, um *passo completo* acontece quando um determinado elemento do vetor é colocado em sua posição definitiva.