

CP3 – Dynamic Programming

Caio Martinez Saes RM: 560753

José Claudio da Silva Junior RM: 559680

Questão 1

Bubble Sort:

```
lista = [38, 12, 44, 27, 9]
aux = 0
contador = 1
contadorTroca = 1
for i in range(len(lista)):
    for j in range(len(lista)):
        if lista[i] < lista[j]:
            aux = lista[i]
            lista[i] = lista[j]
            lista[j] = aux
            print(f"{contadorTroca}º Troca de {lista[i]} por {lista[j]}")
            contadorTroca += 1
    print(f"{contador}º Etapa: {lista}")
    contador += 1
print(f"Total de comparações: {contador}")
print(f"Total de trocas realizadas: {contadorTroca}")
```

Questão 2

Merge Sort

```
aux = 1
def mergeSort(lista):
    global aux
    if len(lista) > 1:
        meio = len(lista)//2

        esq = lista[:meio]
        dir = lista[meio:]
        print(f'esquerda: {esq}, direita: {dir}')

        mergeSort(esq)
        mergeSort(dir)

        i=0
        j=0
        k=0
        while i < len(esq) and j < len(dir):
```

```

        print(f'Etapa: {aux}')

        if esq[i] < dir[j]:
            print(f"Troca: {lista[k]} por {esq[i]}")
            lista[k]=esq[i]
            i=i+1
        else:
            print(f"Troca: {lista[k]} por {dir[j]}")
            lista[k]=dir[j]
            j =j+1
        k=k+1
        aux += 1
    while i < len(esq):
        print(f'Etapa: {aux}')
        print(f"mescla: {esq[i]}")
        lista[k]=esq[i]
        i=i+1
        k=k+1
        aux += 1
    while j < len(dir):
        print(f'Etapa: {aux}')
        print(f"mescla: {dir[j]}")
        lista[k]=dir[j]
        j=j+1
        k=k+1
        aux += 1

lista = [38, 12, 44, 27, 9]
mergeSort(lista)
print(f"Lista final: {lista}")

```

Questão 3

Bubble Sort

Etapa	Ação	Esquerda	Direita	Comparação	Menor inserido	Resultado Parcial
1	Troca de 44 por 38	44	38	44 > 38	38	[38, 12, 44, 27, 9]
2	Troca de 44 por 38	44	38	44 > 38	38	[38, 12, 44, 27, 9]
3	Troca de 44 por 12	44	12	44 > 12	12	[44, 12, 38, 27, 9]
4	Troca de 44 por 12	44	12	44 > 12	12	[44, 12, 38, 27, 9]
5	Troca de 44 por 12	44	12	44 > 12	12	[44, 12, 38, 27, 9]
6	Troca de 44 por 38	44	38	44 > 38	38	[12, 44, 38, 27, 9]
7	Troca de 44 por 38	44	38	44 > 38	38	[12, 44, 38, 27, 9]
8	Troca de 44 por 38	44	38	44 > 38	38	[12, 44, 38, 27, 9]
9	Troca de 44 por 38	44	38	44 > 38	38	[12, 44, 38, 27, 9]
10	Troca de 44 por 38	44	38	44 > 38	38	[12, 44, 38, 27, 9]
11	Troca de 44 por 38	44	38	44 > 38	38	[12, 44, 38, 27, 9]
12	Troca de 38 por 27	38	27	38 > 27	27	[12, 38, 44, 27, 9]
13	Troca de 38 por 27	38	27	38 > 27	27	[12, 38, 44, 27, 9]
14	Troca de 38 por 27	38	27	38 > 27	27	[12, 38, 44, 27, 9]
15	Troca de 38 por 27	38	27	38 > 27	27	[12, 38, 44, 27, 9]
16	Troca de 38 por 27	38	27	38 > 27	27	[12, 38, 44, 27, 9]
17	Troca de 44 por 38	44	38	44 > 38	38	[12, 27, 44, 38, 9]
18	Troca de 12 por 9	12	9	12 > 9	9	[12, 27, 38, 44, 9]
19	Troca de 12 por 9	12	9	12 > 9	9	[12, 27, 38, 44, 9]
20	Troca de 12 por 9	12	9	12 > 9	9	[12, 27, 38, 44, 9]
21	Troca de 27 por 12	27	12	27 > 12	12	[9, 27, 38, 44, 12]
22	Troca de 38 por 27	38	27	38 > 27	27	[9, 12, 38, 44, 27]
23	Troca de 44 por 38	44	38	44 > 38	38	[9, 12, 27, 44, 38]
24	Troca de 44 por 38	44	38	44 > 38	38	[9, 12, 27, 38, 44]
25	Troca de 44 por 38	44	38	44 > 38	38	[9, 12, 27, 38, 44]

Questão 4

Merge Sort

Etapa	Ação	Esquerda	Direita	Comparação	Menor inserido	Resultado Parcial
1	Divisão por 2 (lista principal)	[38, 12]	[44, 27, 9]	/ 2	-	-
2	Troca: 38 por 12	38	12	38 > 12	12	[38]
3	Troca: 27 por 9	27	9	27 > 9	9	[38, 27]
4	Troca: 44 por 9	44	9	44 > 9	9	[44, 38, 27]
5	Troca: 27 por 9	27	9	27 > 9	9	[9, 44, 38, 27]
6	Troca: 38 por 9	38	9	38 > 9	9	[9, 38, 44, 27]
7	Troca: 12 por 12	12	12	12 > 12	12	[12, 9, 38, 44, 27]
8	Troca: 44 por 27	44	27	44 > 27	44	[12, 9, 38, 27, 44]
9	Troca: 27 por 38	27	38	27 < 38	27	[12, 9, 27, 38, 44]
10	Troca: 12 por 9	12	9	12 < 9	9	[9, 12, 27, 38, 44]
11	-	-	-	-	-	[9, 12, 27, 38, 44]

Questão 5

Merge Sort: Possui complexidade de tempo de $O(n \log n)$ tanto no melhor, pior e médio casos. Sendo melhor pois o tempo cresce mais lentamente se comparado a complexidade quadrática.

Bubble Sort: tem complexidade de $O(n^2)$ no pior e caso médio. o melhor caso é $O(N)$

Questão 6

Vantagem Merge Sort: Mais eficiente para listas grandes

Desvantagem Merge Sort : Uso de memoria mais elevado