

lista 05

Questão 1 (bolha, inserção e seleção). Implemente os métodos

```
void bolha(int n, int L[n]){  
    ...  
}
```

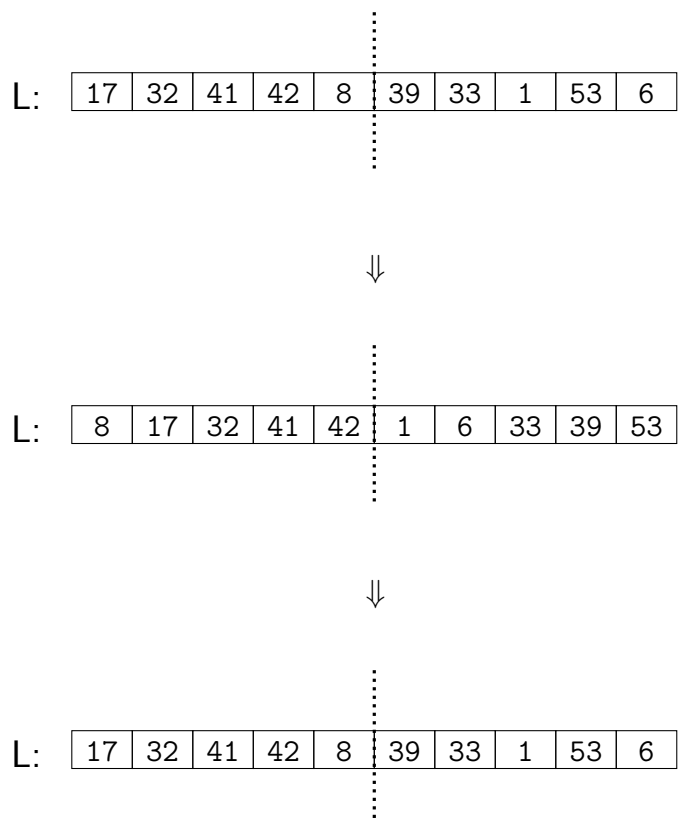
```
void insercao(int n, int L[n]){  
    ...  
}
```

```
void selecao(int n, int L[n]){  
    ...  
}
```

que implementam os algoritmos da bolha, da inserção e da seleção.

Questão 2 (Metade). Considere uma lista L de inteiros positivos de tamanho n . Escreva um programa que realiza as seguintes etapas:

- Ordena a primeira e a segunda metade da lista usando o bolha;
- Faz o merge das duas metades ordenadas em um vetor auxiliar R .



Calcule o número de execuções de instruções (no pior caso) do seu algoritmo. (Se você tiver dificuldade em calcular o número exato, tente fazer as melhores estimativas que você puder.)

Questão 3 (Posições ímpares). Adapte dois algoritmos de ordenação da sua preferência de forma que um deles ordene apenas as posições ímpares e o outro ordene apenas as posições pares de uma lista L dada.

L :

17	32	41	42	8	39	33	1	53	6
----	----	----	----	---	----	----	---	----	---

\Downarrow

L :

8	1	17	6	33	32	41	39	53	42
---	---	----	---	----	----	----	----	----	----

Questão 4 (Cocktail-sort). O algoritmo da bolha pode ser implementado tanto “varrendo” os maiores elementos para o final da lista, como “varrendo” os menores para o começo da lista.

Adapte o algoritmo da bolha para que, a cada iteração, alternadamente seja varrido o maior elemento para a direita e o menor para a esquerda.

Questão 5 (Ordem 3-invertida). Os números naturais podem ser facilmente agrupados de 3-em-3

0 1 2 3 4 5 6 7 8 9 10 11 ...

A ordem 3-invertida inverte esses blocos de tamanho 3

2 1 0 5 4 3 8 7 6 11 10 9 ...

Denotamos a ordem 3-invertida como \prec^3 . Assim, na ordem 3-invertida 8 é menor que 6 ($8 \prec^3 6$), mas 4 continua menor que 10 ($4 \prec^3 10$).

Adapte o algoritmo de ordenação que você menos prefere para, dada uma lista L , ordená-la segundo a ordem 3-invertida.