

Lista de exercícios 3
Estruturas de informação - 2021.2

Questão 1. Crie uma função recursiva para encontrar o maior elemento de um vetor.

Questão 2. Crie uma função recursiva para calcular a soma dos elementos de um vetor.

Questão 3. Crie uma função recursiva para calcular o maior divisor comum entre dois números.

Questão 4. Suponha que temos duas pilhas, P_1 e P_2 , contendo números em ordem crescente (o menor elemento se encontra no topo, e o maior na base da pilha), e queremos combinar os elementos de P_1 e P_2 em uma única pilha, também em ordem crescente. Considere os seguintes algoritmos para essa tarefa:

Algoritmo 1:

```
Enquanto ( $P_1 \neq \emptyset$  e  $P_2 \neq \emptyset$ ) {  
    Comparar os elementos dos topos de  $P_1$  e  $P_2$   
    Mover o menor para a pilha auxiliar  $P_3$ ;  
}  
Mover todos os elementos de  $P_3$  para  $P_1$ ;  
Retornar  $P_1$ ;
```

Algoritmo 2:

```
Mover todos os elementos de  $P_1$  para  $P_3$ ;  
Mover todos os elementos de  $P_2$  para  $P_4$ ;  
Enquanto ( $P_3 \neq \emptyset$  e  $P_4 \neq \emptyset$ ) {  
    Comparar os elementos dos topos de  $P_3$  e  $P_4$   
    Mover o maior para a pilha  $P_1$ ;  
}  
Retornar  $P_1$ ;
```

Observação: Nos algoritmos acima, a comparação só é realizada se as duas pilhas não estão vazias. Caso contrário, o topo da pilha não vazia é selecionado imediatamente.

- (a) Calcule o número de operações (inserções e remoções) realizadas pelos algoritmos 1 e 2, e decida qual deles é mais eficiente.
- (b) Quantas operações são realizadas no Algoritmo 1 no melhor e no pior caso?
- (c) Se as estruturas usadas fossem filas, a tarefa poderia ser realizada em tempo menor? Justifique.

Questão 5. Escreva em pseudo-código uma implementação de duas pilhas em um único vetor de tamanho N de modo que só haja estouro de memória se o tamanho das duas pilhas for maior que N . As operações de inserção e remoção (Push e Pop) de cada pilha devem ser executadas em tempo $O(1)$.

Questão 6. Escreva em pseudo-código uma implementação de um **deque**, que é uma estrutura de dados com duas extremidades, que permite inserir e remover elementos em ambas as extremidades. Sua implementação deve ter as operações Insert e Delete executando em tempo $O(1)$.

Questão 7. Uma palavra é um palíndromo se a sequência de letras que a forma é a mesma, quer seja lida da esquerda para a direita ou da direita para a esquerda (exemplo: raia). Escrever um algoritmo eficiente para reconhecer se uma dada palavra é um palíndromo. Escolher uma estrutura de dados conveniente para representar a palavra (pode ser uma variação/modificação das estruturas vistas em sala).