

Estruturas de Dados
Lista de exercícios 1

1. Prove as seguintes afirmações sobre notação assintótica:
 - $n^3/100 - 25n^2 - 100n + 7$ é $\Theta(n^3)$
 - $77n^3 - 13n^2 + 29n - 5$ é $\Omega(n^3)$
 - $34n \log_7 n^2 + 13n$ é $\Omega(n)$ e $O(n^2)$
2. Considere vetores que satisfazem a propriedade: o subvetor dos índices ímpares está ordenado crescentemente e o dos índices pares está ordenado decrescentemente. Exemplo: $A=[1\ 50\ 2\ 40\ 3\ 30\ 4\ 20\ 5\ 10]$. Faça um algoritmo de tempo $O(\log n)$ que receba um vetor desse tipo e um inteiro x , e informe se x está no vetor, retornando sua posição, se for o caso.
3. Reescreva os algoritmos InsertionSort, BubbleSort e HeapSort em pseudo-código de modo que receba como entrada um vetor A e índices $p < r$ e ordene o subvetor $A[p, \dots, r]$.
4. O algoritmo HeapSort visto em sala ordena um vetor $A[1, \dots, N]$ usando um heap máximo enraizado na posição 1 do vetor A . Reescreva esse algoritmo em pseudo-código de modo a utilizar um heap **mínimo** enraizado na última posição do vetor A .
5. Escreva em pseudo-código um algoritmo não-recursivo de tempo $O(n)$ que inverta uma lista simplesmente ligada de n elementos, sem criar mais memória.
6. 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 Push e Pop de cada pilha devem ser executadas em tempo $O(1)$.
7. Escreva em pseudo-código um algoritmo que receba uma expressão em notação polonesa reversa e a converta para a notação bem parentizada, usando uma pilha ou uma fila. Faça o mesmo da notação polonesa para a notação bem parentizada.
8. 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 $O(1)$. Faça uma implementação usando vetores e faça outra implementação sem usar vetores.
9. Dados inteiros N e $k < N$, o Problema de Josephus consiste no seguinte: N homens estão em um círculo e começam a se matar de k em k . Ou seja, começando do homem número 1, $k - 1$ homens vivos são pulados e o próximo é morto, e assim sucessivamente, até que reste apenas $k - 1$ homens vivos. Por exemplo, para $k = 2$, se $N = 2, 3, 4, 5, 6$, os sobreviventes são respectivamente os homens número 1, 3, 1, 3 e 5. Faça um algoritmo usando uma lista circular que, recebendo como entrada inteiros N e k , retorne os sobreviventes.