



IMD0029 – ESTRUTURAS DE DADOS BÁSICAS I
PROF. EIJI ADACHI M. BARBOSA

Lista de Exercícios – Busca Binária

Questão 1. Dado um inteiro X e um array A contendo N inteiros distintos, o problema *Soma-2* é definido como o problema de determinar se existem dois índices i e j tais que $A[i] + A[j] == X$. Neste contexto, projete duas soluções para o problema *Soma-2* com complexidades $\Theta(n^2)$ e $\Theta(n \cdot \lg(n))$.

Questão 2: Considere um array A contendo N inteiros, com possíveis repetições, já ordenado. Neste contexto:
(A) Projete um algoritmo que recebe como entrada um inteiro K e retorna um índice i tal que $A[i] == K$, sendo $A[i]$ o elemento igual a K que está mais à esquerda em A . Sua solução deverá ter complexidade $\Theta(\lg(n))$.
(B) Projete um algoritmo que recebe como entrada um inteiro K e retorna quantos elementos em A tem valor igual a K ocorrem no array A . Sua solução deverá ter complexidade $\Theta(\lg(n))$.

Questão 3: Projete um algoritmo de busca ternária, isto é, um algoritmo de busca que divide o seu espaço de busca em três. Faça duas versões da busca ternária: uma iterativa e outra recursiva. Qual a complexidade do algoritmo de busca ternária?

Questão 4: Sequências bitônicas são aquelas que possuem duas sequências, sendo uma sequência inicial crescente, seguida de uma sequência decrescente. Ou seja, os elementos de uma sequência bitônica obedecem a seguinte relação:

$$A_0 < \dots < A_{i-1} < A_i > A_{i+1} > \dots > A_n$$

Considere que um array bitônico é um array de inteiros sem repetições cujos elementos representam uma sequência bitônica. Dado este contexto:

- A) Projete um algoritmo que recebe como entrada um array de inteiros e verifica se o array é bitônico ou não.
- B) Projete um algoritmo que recebe como entrada um array bitônico e retorna o índice do pico, isto é, o elemento A_i . Sua solução deverá ter complexidade $\Theta(\lg(n))$.
- C) Projete um algoritmo que recebe como entrada um array bitônico A e um número inteiro K e retorna o índice i tal que $A[i] == K$. Sua solução deverá ter complexidade $\Theta(\lg(n))$.

Questão 5. Considere que um array inicialmente ordenado foi deslocado à direita sem querer e não se sabe quantas vezes ele foi deslocado à direita. Um deslocamento à direita de um array faz com que o último elemento do array torne-se o primeiro. Por exemplo: dado o array de entrada $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, após 3 deslocamentos à direita, o array estará da seguinte forma: $\{8, 9, 10, 1, 2, 3, 4, 5, 6, 7\}$. Neste contexto:

- A) Projete um algoritmo que recebe como entrada um array A que foi deslocado à direita e retorna o número de vezes que este array foi deslocado à direita. Sua solução deverá ter complexidade $\Theta(\lg(n))$.
- B) Projete um algoritmo que recebe como entrada um array A que foi deslocado à direita e um inteiro K e retorna o índice i tal que $A[i] == K$. Sua solução deverá ter complexidade $\Theta(\lg(n))$.