

Algoritmos e Estruturas de Dados II

Primeira Lista de Exercícios – TADs e Análise de Complexidade

TADs

1. O que é um Tipo Abstrato de Dados (TAD) e qual a característica fundamental na sua utilização?
2. Quais as vantagens de se programar com TADs?
3. Crie um Tipo Abstrato de Dados (TAD) que represente o tipo conjunto de inteiros, utilizando uma representação de vetor de inteiros e que contenha as seguintes funções:
 - a) União
 - b) Cria um conjunto vazio
 - c) Insere
 - d) Remove
 - e) Intersecção
 - f) Diferença
 - g) Testa se um número pertence ao conjunto
 - h) Menor valor
 - i) Maior valor
 - j) Testa se os conjuntos são iguais
 - k) Verifica o tamanho do conjunto
 - l) Testa se o conjunto é vazio
4. Crie um Tipo Abstrato de Dados (TAD) que represente os números racionais e que contenha as seguintes funções:
 - a) Cria racional
 - b) Soma racionais
 - c) Multiplica racionais
 - d) Testa se são iguais
5. Desenvolva um TAD para um cubo. Inclua as funções de inicializações necessárias e as operações que retornem os tamanhos de cada lado, a sua área e o seu volume.
6. Desenvolva um TAD para um cilindro. Inclua as funções de inicializações necessárias e as operações que retornem sua altura e raio, a sua área e o seu volume.
7. Desenvolva um TAD para uma esfera. Inclua as funções de inicializações necessárias e as operações que retornem seu raio, a sua área e o seu volume.

Análise de Complexidade

1. O que significa dizer que uma função $g(n)$ é $O(f(n))$?
2. Explique a diferença entre $O(1)$ e $O(2)$.
3. Indique se as afirmativas a seguir são verdadeiras ou falsas:
 - (a) $2^{n+1} = O(2^n)$
 - (b) $2^{2n} = O(2^n)$
 - (c) $f(n) = O(u(n))$ e $g(n) = O(v(n)) \rightarrow f(n) + g(n) = O(u(n) + v(n))$
4. Se os algoritmos A e B levam tempo $a(n) = n^2 - n + 549$ e $b(n) = 49n + 49$.
 - a) $a(n) = O(b(n))$?
 - b) $b(n) = O(a(n))$?
5. Considere um algoritmo para inserir um elemento em um arranjo ordenado de elementos.
 - a) Qual o número mínimo de passos para resolver este problema?
 - b) Qual o melhor caso?
 - c) Qual o pior caso?
 - d) Qual o caso médio?
6. Se os algoritmos A e B levam tempo $a(n) = n^2 - 2 + 549$ e $b(n) = n^2 + 30n$.
 - a) $a(n) = O(b(n))$?
 - b) $b(n) = O(a(n))$?
 - c) Estes dois algoritmos em algum momento (dependente do tamanho de n) irão dar o mesmo resultado? Se Sim para qual o valor de n ?
 - d) Para quais valores A leva menos tempo para executar do que B?
7. Qual a ordem de complexidade das sentenças abaixo:
 - a) $f(n) = 4n + n - 2 + 3$
 - b) $g(n) = 4n^2 + 3n^3 + 2n - 2$
 - c) $h(n) = 2^{2n} + 4n^3$
 - d) $i(n) = 2^{5n} + 4n^2$
8. Faça um método que receba um número inteiro n e efetue o número de multiplicações, pedido nos casos a seguir:
 - a) $5n + 4n^3$
 - b) $9n^4 + 5n^2 + n/2$

- c) $4n^3 + 2$
- d) $\lg(n) + n^2$
- e) $3\lg(n) + \lg(n)$
- f) $2n + 2n^2 + \lg(n)$

9. Apresente a função e a taxa de complexidade para as 3 notações vistas em sala, referente ao número de comparações, para o pior e melhor caso, para as opções a seguir:

a)

```
void imprimirMaxMin(int [] array, int n){
    int maximo, minimo;

    if (array[0] > array[1]){
        maximo = array[0];    minimo = array[1];
    } else {
        maximo = array[1];    minimo = array[0];
    }

    for (int i = 2; i < n; i++){
        if (array[i] > maximo){
            maximo = array[i];
        } else if (array[i] < minimo){
            minimo = array[i];
        }
    }
}
```

b)

```
i = 0;

while (i < n) {
    i++;
    a--;
}

if (b > c) {
    i--;
} else {
    i--;
    a--;
}
```

c)

```

for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
        a--;
        b--;
    }
    c--;
}

```

10 – Apresente o tipo de crescimento que melhor caracteriza as funções abaixo:

	Constante	Linear	Polinomial	Exponencial
$3n$				
1				
$(3/2)n$				
$2n^3$				
2^n				
$3n^2$				
1000				
$(3/2)^n$				

11 – Classifique as funções $f_1(n) = n \cdot \lg(n)$, $f_2(n) = \lg(n)$, $f_3(n) = 8n^2$, $f_4(n) = 64$, $f_5(n) = 6n^3$, $f_6(n) = 8^{2n}$ e $f_7(n) = 4n$ de acordo com o crescimento, do mais rápido para o mais lento.