PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Disciplina: Algoritmos e Estruturas de Dados

Nome: Linekker Emmanuel Batista Melo

Lista de Exercícios 4

1) Consultar os slides.
2) Apresente a ***função de complexidade*** e a ***complexidade assintótica*** para o número de movimentações de registros para o pior e melhor caso (apresente a prova da complexidade assintótica).

```
public static 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];
            }
      }
}
```

Resposta:
Função de Complexidade:
Pior caso: f(n) = 2 + (n-2) (O pior caso ocorre quando em todas as iterações do *for* um teste for verdadeiro, seja o teste do *if* ou do *else if*. Exemplo [1 2 3 4 5 6 7])
Melhor caso: f(n) = 2 (O *melhor caso* ocorre quando o maior e o menor elementos estiverem nas duas posições iniciais do vetor, assim dentro do *for* todos os testes serão falsos e nenhuma movimentação será realizada dentro do *for*. Exemplo: [1 10 3 2 5 9])

Complexidade assintótica:
Pior caso: O(n) (Prova f(n) = 2 + (n-2)= n é O(n), quando c = 1 e m = 1)
Melhor caso: O(1) (Prova f(n) = 2 é O(1), quando c = 2 e m = 1)

3) Apresente a ***função de complexidade*** e a ***complexidade assintótica*** para o número de multiplicações para o pior e melhor caso (apresente a prova da complexidade assintótica).

```
public static int alg1(int n){
      int res = 1;
      for(int i=n; i>1; i--){
            res = res*i;
      }
      return res;
}
```

Resposta:
Função de Complexidade:
Pior e melhor caso: f(n) = n-1

Complexidade assintótica:
Pior e melhor caso: O(n) (Prova f(n) = n-1 é O(n), quando c = 1 e m = 1)

4) Apresente a ***função de complexidade*** e a **complexidade assintótica** para o número de subtrações para o pior e melhor caso (apresente a prova da complexidade assintótica).

```
…
i = 0;
while (i < n) {
       i++;
       a--;
}
if (b > c) {
       i--;
} else {
       i--;
       a--;
}
```

Resposta:

Função de Complexidade:

Pior caso: $f(n) = n+2$ (O pior caso ocorre quando o *if* for falso, pois assim serão executadas as duas subtrações dentro do *else*)

Melhor caso: $f(n) = n+1$ (O melhor caso ocorre quando o *if* for verdadeiro, pois assim será executada a subtração dentro do *if*)

Complexidade assintótica:

Pior caso: $O(n)$ (Prova $f(n) = n+2$ é $O(n)$, quando $c = 3$ e $m = 1$)

Melhor caso: $O(n)$ (Prova $f(n) = n+1$ é $O(n)$, quando $c = 2$ e $m = 1$)

5) Prove as afirmações abaixo:

a) $7n-2$ é $O(n)$

Prova: pela definição da notação O, precisamos achar uma constante *c* positiva e uma constante *m* positiva, tal que $7n-2 <= cn$ para todo inteiro $n >= m$. Uma escolha poderia ser $c = 7$ e $m = 1$. De fato, esta é uma das infinitas escolhas possíveis, porque qualquer número real maior ou igual a 7 será uma escolha possível para *c*, e qualquer inteiro maior ou igual a 1 é uma escolha possível para *m*.

b) $20n^3+10n\log(n)+5$ é $O(n^3)$

Prova: pela definição da notação O, precisamos achar uma constante c positiva e uma constante m positiva, tal que $20n^3+10n\log(n)+5 <= cn^3$, para todo inteiro $n >= m$. Uma escolha poderia ser $c = 35$ e $m = 1$ (esta é uma das infinitas escolhas possíveis para as constantes *c* e *m*)

c) 2^{100} é $O(1)$

Prova: pela definição da notação O, precisamos achar uma constante c positiva e uma constante m positiva, tal que $2^{100} <= c(1)$, para todo inteiro $n >= m$. Uma escolha poderia ser $c = 2^{100}$ e $m = 1$ (esta é uma das infinitas escolhas possíveis para as constantes *c* e *m*)

d) $5/n$ é $O(1/n)$

Prova: pela definição da notação O, precisamos achar uma constante c positiva e uma constante m positiva, tal que $5/n <= c(1/n)$, para todo inteiro $n >= m$. Uma escolha poderia ser $c = 5$ e $m = 1$ (esta é uma das infinitas escolhas possíveis para as constantes *c* e *m*)

6) Indique se a afirmativa a seguir é falsa ou verdadeira e justifique sua resposta:

$$n^2 + 10^{10^{100}} n + 123 = O(n).$$

A afirmação é falsa.

Para essa afirmação ser verdadeira, de acordo com a definição da notação O, devem existir constante positivas c e m, tais que

$$n^2 + 10^{10^{100}} n + 123 \leq cn$$

para todo $n >= m$.

Ao dividirmos cada termo dessa inequação por n, temos

$$n + 10^{10^{100}} + \frac{123}{n} \leq c$$

Não existe constante positiva c que seja limitada pelo termo da esquerda da inequação à medida que *n* cresce. Assim, $n^2 + 10^{10^{100}} n + 123 \neq O(n)$

7) Apresente a ***função de complexidade*** e a **complexidade assintótica** para o número de subtrações para o pior e melhor caso (apresente a prova da complexidade assintótica).

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

Resposta:
Função de Complexidade:
Pior e melhor: f(n) = (2n+1)n

Complexidade assintótica:
Pior e melhor: $O(n^2)$ (Prova f(n) = (2n+1)n = $2n^2$+n é $O(n^2)$, quando c = 3 e m = 1)

Nem sempre o valor da constante *m* poderá ser 1. O valor da constante *m* vai depender do valor escolhido para a constante *c*. Por exemplo:

f(n) = n + 7 é O(n)?

Prova: pela definição da notação O, precisamos achar uma constante c positiva e uma constante m positiva, tal que n + 7 <= cn para todo inteiro n>= m.

Uma escolha poderia ser c = 2. Para esse valor de *c*, o valor de *m* tem que ser maior ou igual a 7. Uma vez que, para valores menores a inequação seria falsa.

Exemplos:

(1) + 7 <= 2(1)
8 <= 2 FALSO

(3) + 7 <= 2(3)
10 <= 6 FALSO