

Universidade Federal de Goiás
Estruturas de Dados II
Prof. Jesus José de Oliveira Neto

Exercício 01 – Faça a análise dos algoritmos abaixo:

- a) int Algoritmo1(int n) { {Número de vezes}
1 int i = 1;
2 int x = 1;
3 while (i <= n) {
4 x = x * i;
5 i = i + 1;
6 }
7 return x;
8 }
- b) int Algoritmo2(int n) { {Número de vezes}
1 int i = n;
2 int y = 0;
3 while (i >= 1) {
4 y = y + i;
5 i = i - 1;
6 }
7 return y;
8 }
- c) float Algoritmo3(int n, float A[]) {Número de vezes}
1 int k = n - 1;
2 float y = 0.0;
3 while (k >= 0) {
4 y += A[k] ;
5 k = k - 1;
6 }
7 return y;
8 }
- d) float Algoritmo4(int n, float A[]) {Número de vezes}
1 float z = A[0];
2 int i = 1;
3 while (i <= n - 1) {
4 if(A[i] < z) {
5 z = A[i];
6 }
7 i = i + 1;
8 }
9 return z;
10 }

d) `int Algoritmo5(int n, int A[])` {Número de vezes}

```
1  int s = 0;
2  int i = 0;
3  while (i < n) {
4      if(A[i] % 2 == 0) {
5          s = s + A[i];
6      }
7      i = i + 1;
8  }
9  return s;
10 }
```

e) `int Algoritmo6(int mat[][], int n) { // matriz nxn` {Número de vezes}

```
1  int i, j;
2  int temp = 0;
3  i = 0;
4  while(i < n) {
5      j = 0;
6      while(j < n) {
7          temp += mat[i][j];
8          j++;
9      }
10     i++;
11 }
12 return temp;
13 }
```

Exercício 02 – Implemente o código de cada um dos algoritmos acima e verifique o tempo de execução de cada um deles para grandes valores de **n**. Utilize o método *System.currentTimeMillis()*.