Avaliação 1 – Complexidade de Algoritmos

Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ RA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Cada item de questão vale 1 ponto, salvo anotação em contrário.

Seja o algoritmo definido pelo método abaixo:

public int[] void minMax(int[] vi, n) {

int min, max;

min = max = vi[0];

for (int i=1; i < n; i++) {

if (vi[i] > max) max = vi[i];

if (vi[i] < min) min = vi[i];

}

int[] res = new int[2];

res[0] = min;

res[1] = max;

}

Q1. Especifique o problema que o algoritmo procura resolver:

1. de maneira informal: explique verbalmente o que o problema.
2. de maneira formal, como feito na transparência 11 do arquivo algo1.pdf.

Q2. Calcule:

1. quantas vezes as instruções do laço são executadas?
2. quantas vezes as condições para vi[i] são calculadas na execução do algoritmo?
3. quantas vezes a condição (i< n) é calculada?

Q3. Seja o polinômio: P(x) = 10 x4 + 3 x3 + 2 x2 + 5 x – 4, analise a complexidade (tempo de execução) de cada um dos algoritmos abaixo, supondo que o tempo de uma adição é menor do que o tempo de uma multiplicação que é menor do que o tempo para calcular a função pow().

double poli1(double x) {

double coeficientes[] = {-4., 5., 2., 3., 10.};

int i, j;

double res = 0.;

for (i = 0; i < 5; i++) {

double termo = coeficientes[i];

for (j = 0; j < i; j++) termo \*= x;

res += termo;

}

return res;

}

double poli2(double x) { // P(x)=-4 + x(5 + x(2 + x(3 + 10x)))

double coeficientes[] = {-4., 5., 2., 3., 10.};

int i;

double res = coeficientes[4];

for (i = 4; i >0; i--) {

res = coeficientes[i-1] + res \* x;

}

return res;

}

double poli3(double x) {

double coeficientes[] = {-4., 5., 2., 3., 10.};

int i;

double res = 0.;

for (i = 4; i >= 0; i--) {

res += coeficientes[i] \* pow(x, i);

}

return res;

}

Q4. Qual dos algoritmos acima é o mais rápido?

Q5. Qual a complexidade assintótica de um algoritmo que tenha de calcular a média dos valores de um vetor de n elementos? Forneça o algoritmo para este cálculo.