Hélder Silva Ferreira Lima

2 Determine Θ for the following code fragments in the average case. Assume

that all variables are of type int.

(a)

a = b + c; // O(1)

d = a + e; // O(1)

// F = c1 + c2 = c3

// O(1)

(b)

sum = 0;

for (i=0; i<3; i++) //3

for (j=0; j<n; j++) n

sum++; //c

// F = 3\*n\*c = 3cn

// O(n)

(c)

sum=0; //c

for (i=0; i<n\*n; i++) //n\*n;

sum++; //c

//O(n^2)

(d)

for (i=0; i < n-1; i++) //n-1

for (j=i+1; j < n; j++) { // no pior caso, n

tmp = A[i][j]; //c

A[i][j] = A[j][i]; //c

A[j][i] = tmp; //c

}

// F = (n-1) \* (n\*3c) = (n-1)\*n = n^2-n = n^2

// O(n^2)

(e)

sum = 0; //c

for (i=1; i<=n; i++) //n

for (j=1; j<=n; j\*=2) //log(n)

sum++; //c

//F = c + n\* (log(n) \* c) = c + cnlog(n) = nlog(n)

//O(nlog(n))

(f)

sum = 0; //c

for (i=1; i<=n; i\*=2) //log(n)

for (j=1; j<=n; j++) //n

sum++; //c

//F = c + log(n) \* (n\*c) = c + cnlog(n) = nlog(n)

//O(nlog(n))

(g)

Assume that array A contains n values, Random takes constant time,

and sort takes n log n steps.

for (i=0; i<n; i++) { //n

for (j=0; j<n; j++) //n

A[j] = Random(n); // c

sort(A, n); //nlog(n)

}

// n log n é executado n vezes -> n \* nlogn

// n do for com j é executado n vezes -> n\*n

// F = n \* [n + nlogn] = n^2 + n^2logn

// O(n^2log(n))

(h)

Assume array A contains a random permutation of the values from 0 to n − 1.

sum = 0; //c

for (i=0; i<n; i++) //n

for (j=0; A[j]!=i; j++) //no pior caso, ele percorre de 0 até n-1 -> n

sum++; // c

// F = n \* n = n^2

// O(n^2)

(i)

sum = 0; //c

if (EVEN(n)) //c

for (i=0; i<n; i++) //n

sum++; //c

else

sum = sum + n; //c

//o pior caso é EVEN(n) ser true, quando F = n.

// O(n)