INSTITUTE OF COMPUTING - UNICAMP

Graduate Program

MO417A Design and Analysis of Algorithms

2015 - Semester 1 - Jorge Stolfi

Problem Set 01 - 2015-03-09

Nome	**RA**

Item													TOT
Nota													

1. Consider the two sorting algorithms below, *insertion sort* I and *selection sort* S. Both algorithms sort a list of n numbers $x_0, x_1, \ldots, x_{n-1}$ in increasing order. The notation $[v]$ before a statement means "v is the number of times that this statement gets executed when the algorithm is executed once".

```
        Algorithm I(n, x)
          for i from 1 to n − 1 do
[f_I]       t ← x_i;
            j ← i;
            while j > 0 and x_{j−1} > t do
[g_I]         x_j ← x_{j−1};
              j ← j − 1;
            x_j ← t;
```

```
        Algorithm S(n, x)
          for i from 0 to n − 2 do
[f_S]       k ← i;
            for j from i + 1 to n − 1 do
[g_S]         if x_j < x_k then
[h_S]           k ← j;
            swap x_i, x_k;
```

(a) Write formulas for the counting variables f_I, f_S, g_I, g_S, and h_S, as a function of n, for If you cannot find exact formulas, or the value depends on the list x, find the best lower and upper bounds that you can.

answer

(b) What can you say about the relative efficiency of the two algorithms?

answer

2. This question refers to algorithm H below, that takes as inputs a natural number n, and a list X of n integers, and returns an integer m. (The algorithm does not solve any interesting problem, it is just an exercise in analysis.)

```
           Algorithm H(n, X)
[f1]           m ← 0; i ← 0;
               while
[f2]               i < n
               do
[f3]               j ← i;
                   while
[f4]                   j < n
                   do
[f5]                   A ← 0;
[f6]                   r ← i;
                       while
[f7]                       r ≤ j
                       do
[f8]                       A ← 2A + X_r;
[f9]                       r ← r + 1;
                       endwhile;
[f10]                  if A = 0 then
[f11]                      m ← m + 1;
                       endif;
[f12]                  j ← j + 1;
                   endwhile;
[f13]              i ← i + 1;
               endwhile;
[f14]          return m;
```

(a) Write the Kirchoff laws that relate the counting variables f_1, f_2, ... (Drawing a flowchart of the algorithm may help.)

answer

(b) Identify a "key" subset of the counting variables f_1, f_2, ..., such that any other counting variable can be computed from that set, using the Kirchoff laws.

answer

(c) Give formulas, as a function of n, for those key variables. If some variable depends on the matrix X, not only on n, give the best upper and lower bounds for that variable that you can find.

answer