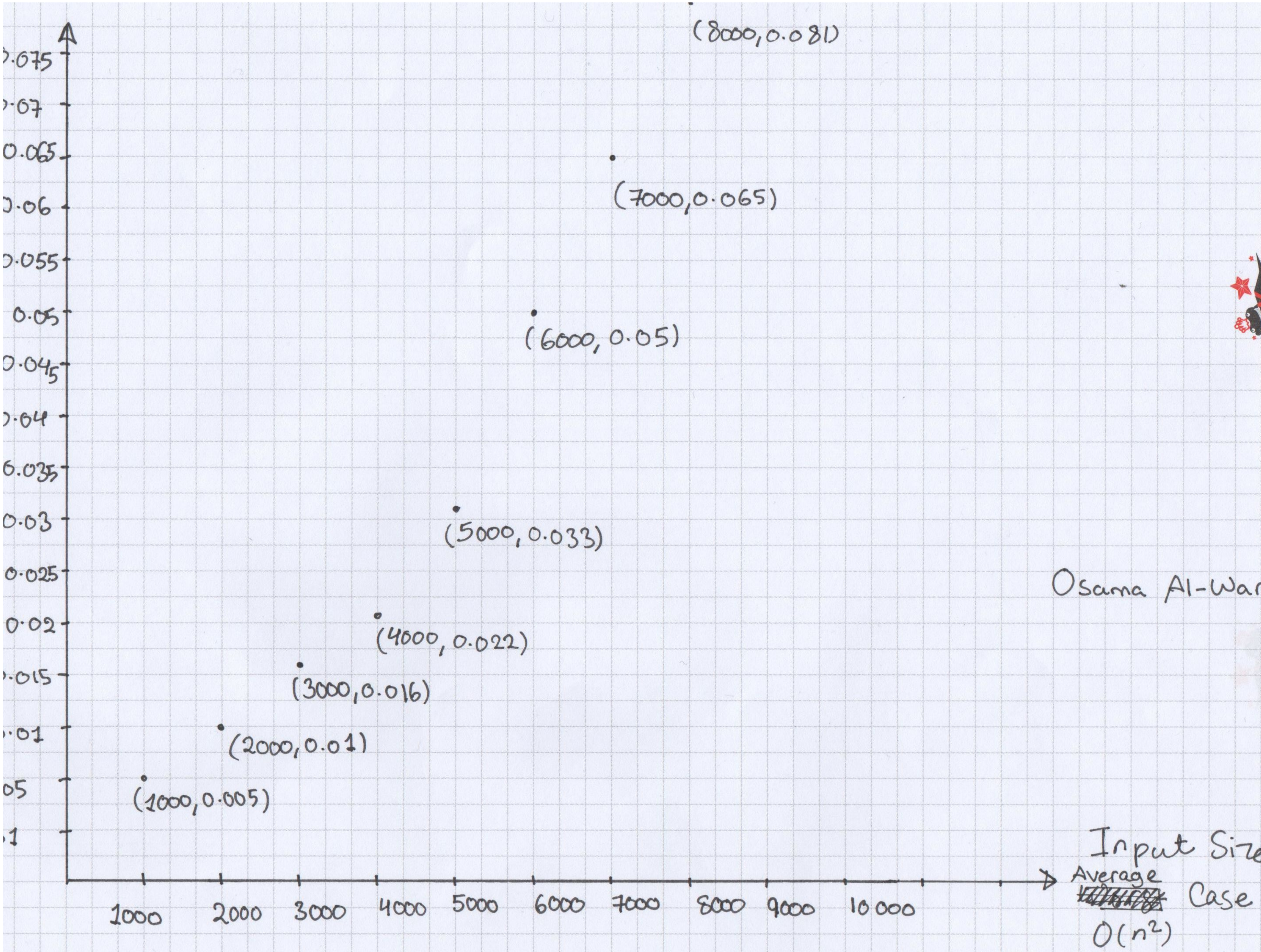
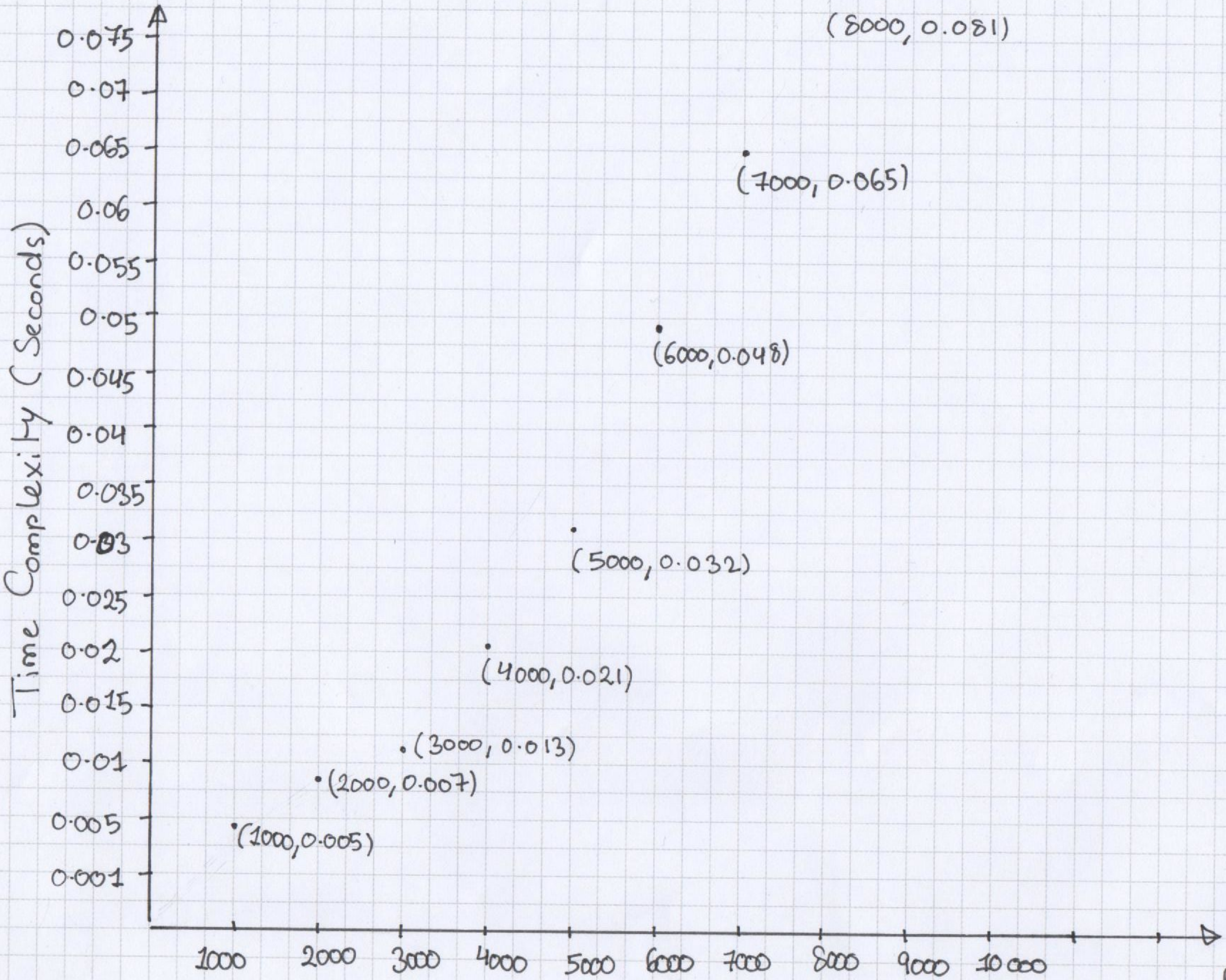


Osama Al-Wardi



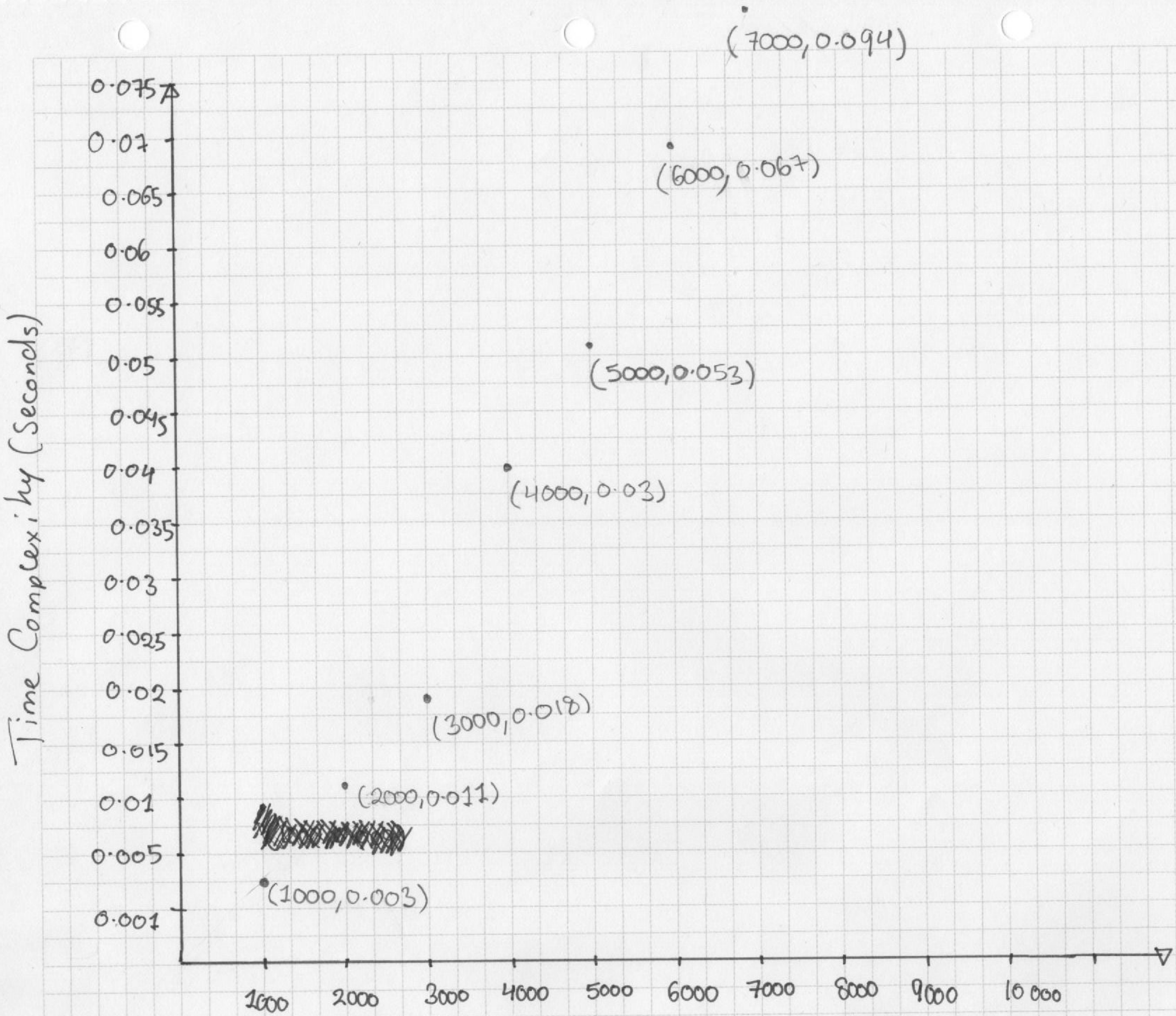
Osama Al-Wardi

Input Size(n)

$\Omega(n^2)$

Best Case $O(n^2)$

Sorted Input



Osama Al-Wardi

Input Size (n)

Worst Case

$O(n^2)$

Invéteel Input

Decending Order

Problem 2

b) Prove Correctness of Selection Sort

Loop Invariant: Before the start of each loop arr[min] <= arr[i...j-1].

First Iteration or Base Case: Lets consider the Subarray of Sorted numbers arr[i...j-1]. Since j=i+1 and i=0 in the first iteration So j=1.

Therefore our Statement (int min = arr[i];) is true because we only have one element in the sorted Subarray. So our loop Invariant is True.

Maintenance or Induction Step:
During iteration j we have an if Statement with two Cases (if A[j]< min) if the Statement is false nothing is excuted. But if it's true it's going to swap A[j] with the position of min and so putting A[j] in the Sorted Subarray. This process is repeated n times, So that the whole array is sorted.

Termination or Final Iteration:

At the last iteration min indexes the last element of the array at position (arr[n];]. Also j=n+1 which means that we will not enter the Second for loop. So the last element in the unsorted Subarray [j...n] is going to be the largest and so it is going to be left at the end which is the correct Location, leaving the whole array Sorted.

This Confirms our loop Invariant and therefore the Algorithm is correct.

Osama Al-Wardi
CS Problem Sheet 1
12th / Feb

Problem 1

a) $f(n) \in O(g(n))$

$g(n) \in \Omega(f(n))$

$f(n) \in o((g(n))$

$g(n) \in \omega(f(n))$

$g(n) \in (c\, f(n))$

b) $g(n) \in O(f(n))$

$f(n) \in \Omega(g(n))$

$f(n) \in \Theta(f(n))$

$f(n) \in o(g(n))$

$f(n) \in \omega(g(n))$

c) $g(n) \in O(f(n))$

$f(n) \in \Omega(g(n))$

$g(n) \in o(f(n))$

$f(n) \in \omega(g(n))$ $\quad$ $f(n) \in \Theta(g(n))$

d) $f(n) \in \Omega(g(n))$

$g(n) \in O(f(n))$

$g(n) \in o(f(n))$

$f(n) \in \omega(g(n))$

$f(n) \in \Theta(g(n))$