**CENG502 Algorithm Analysis and Complexity Theory**

**Assignment 1**

**Given:** Wed, 27 Mar **Due date:** Sat, 6 Apr

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Problem** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Total** |
| **Earned** |  |  |  |  |  |  |  |  |  |
| **Max** | **6** | **5** | **5** | **5** | **13** | **8** | **7** | **5** | **50 + 6 bonus** |

**Problem 1.** [6=2@3 marks] Calculate:

1. .
2. .

**Problem 2.** [5 Marks] Program P run on computer A solves an instance of size n in 10-4 \* n2  milliseconds. If you can get a machine B that runs 1000 times faster, what size of n will you be able to solve in the same time that computer A needed to solve the instance of size 20?

Hint: For a computer, tn = cn . Δ where cn is the number of elementary operations in a program, and Δ is the execution time for one elementary operation. A computer which is 100 times faster will execute an elementary operation for Δ /100 sec.

**Problem 3.** [5 marks] A program with running time takes 2 ms for an input of size 100.

How much time will be needed for this program to solve an instance of size 2500? In general, how many times the running time will increase when the size is increased 25 times? Show your calculations.

Hint: Use that if tn = Θ(f(n)) then we can write tn = cf(n).

**Problem 4.** [5 marks] Indicate, for the pair of expressions <A, B> in the table below, whether A is , , , , or  of B (write "Yes" or "No" in the corresponding box). Show your work. Hint: you may want to use the limit rule or the definitions of the asymptotic notations.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B |  |  |  |  |  |
| 5.(3n).105 | (2.5)n+2 |  |  |  |  |  |
| 3n | n\*2n |  |  |  |  |  |

**Problem 5**. Consider the following algorithm (in Python)

*def foo(n):*

*q = 0*

*for j in range n:*

*q = q + (j \* j) + j*

*return q*

1. [3 marks] Find the exact number of arithmetic operations (the total of additions and multiplications) performed by this algorithm, as a function of n.
2. [1 mark] Determine the order of the running time.
3. [3 marks] What does the algorithm calculate? (Express q as a function of n).
4. [3 marks] Write another Python function to solve the same problem in constant time. Call this program *foo2*.Hint: use the result of the previous task.
5. [3 marks] Run *foo* and *foo2* for values of the argument 25 and 100. Print the results. Include a screenshot.

**Problem 6**.

1. [3 marks] Compare the functions and . Prove your result.
2. [2 marks] Find the order of tn = 0.0555+5005. Explain.
3. [3 marks] Compare the functions and . Prove your result.

**Problem 7**. The following Python program is given:

*def mystery(n):*

*r=0*

*for i in range(n):*

*i+=1*

*for j in range(n,i,-1):*

*for k in range(1,i+1):*

*r=r+1 #(A)*

*return r*

1. [5 marks] Find the exact formula for the number of additions in line (A).
2. [1 mark] Find the asymptotic order for the number of additions using simplified analysis.
3. [1 mark] Compare the results of tasks a) and b).

**Problem 8**. [5 marks] Consider again the element uniqueness problem from Lecture 5d (Example 1). Find the number of comparisons (precise formula and order of magnitude) in the average case under the following assumption:

The array has n cells where n is even. It is known that n/2 different numbers (e.g., 1, 2, 3,…, n/2), are stored in these n cells. Each number is stored twice (there are two numbers 1 in the array, two numbers 2, and so on).

**Example:** n=10, the numbers are {1,2,3,4,5}  
  
One possible instance is [3,2,5,3,4,4,2,1,1,5]. Any of the numbers can be in position 0 - in our case this is element 3. Then, the second number 3 can be anywhere in positions 1,2,...9, in our case it is in position 3.  
  
Submission: Write your answers on a computer, convert into a single pdf file and submit through the link on the course website by the due date. The usual early/late submission policy applies.