CMPE 300 PROJECT 1

Consider the following algorithm which takes a list arr[0: $n-1$] as input. Suppose that n=2^k -1 for some k. Each element in the list is 0, 1 or 2; i.e. arr[i] $\in$ {0,1,2}, $0 \le i \le n-1$. For each i, the probability that arr[i] = 0 is 1/3, the probability that arr[i] = 1 is 1/3 and the probability that arr[i] = 2 is 1/3 , $0 \le i \le n-1$

```
function Example (arr[0:n-1])
Input: arr[0:n-1] (a list of size n)
Output: arr2[0:4] (a list of size 5)

arr2 ← [0,0,0,0,0]
for i ← 0 to n-1 do
   if arr[i] = 0 then                                  (1)
       for t1 ← i to n-1 do
           p1 ← t1^1/2                                 (5)
           x1 ← n+1
           while x1 >= 1 do
               x1 ← ⌊x1/2⌋
               arr2[i%5] ← arr2[i%5] + 1               (2)
           endwhile
       endfor
   else if arr[i] = 1 then
       for t2 ← n downto 1 do
           for p2 ← 1 to n do                          (4)
               x2 ← n+1
               while x2 > 0 do
                   x2 ← ⌊x2/2⌋
                   arr2[i%5] ← arr2[i%5] + 1           (2)(3)
               endwhile
           endfor
       endfor
   else if arr[i] = 2 then
       for t3 ← 1 to n do                      (4)
           x3 ← t3 + 1
           for p3 ← 0 to t3^2 - 1 do
               arr2[i%5] ← arr2[i%5] + 1               (2)(3)
           endfor
       endfor
   endif
endfor
return arr2
```

Theoretical analysis

Consider each of the following five cases separately (one by one):

i) Basic operation is the comparison marked as (1)

ii) Basic operations are the three assignments marked as (2)

iii) Basic operations are the two assignments marked as (3)

iv) Basic operations are the two loop incrementations marked as (4)

v) Basic operation is the assignment marked as (5)

For each case:

a) Analyze B(n)
b) Analyze W(n)
c) Analyze A(n)

Your analyses must be exact. That is, for each analysis, first find the “exact” number of basic operations and then convert them to asymptotic notation.
(Note that best-case input and worst-case input may be different for different cases.) So, you will obtain 15 analysis results.

Real Execution

Then, identify the “correct” basic operation(s) – i.e. the operation that characterizes this algorithm.

Code this algorithm in a programming language you wish. Execute it on a computer with 10 different input sizes n {1, 5, 10, 25, 50, 75, 100, 150, 200, 250}, for three types of inputs (best-case input, worst-case input, average-case input). For each, record the actual execution time (in milliseconds, seconds, etc.). At the end, you should have 30 different time records. Then, you will fill the table in the answer sheets according to the results you get. Also, your code should print out the results. For more information, see section Code.

As an average-case input, you can generate a random array arr[0:n-1] formed of 0s, 1s and 2s according to the probability distribution explained at the beginning. Note that in order to observe the average behavior, for each input size, you must execute the algorithm with random inputs several times and take the average. For this project, run the algorithm at least 3 times for taking the average.

Comparison of theoretical analysis and real behavior

Compare the theoretical results and the actual execution times using graphs. Use graphs where the x-axis denotes the input size and the y-axis denotes the complexity/time. Your comparisons must include all the theoretical and actual analyses you derived above. For each of the 15 theoretical analyses, comment on the result of the comparison with the related actual execution time.

Code

You will also submit your code. You are free to select any language. (Python is preferred.) Yet, please provide a ReadMe file that simply explains how to run your code.

Then you should print the time elapsed for best, worst and average cases, and for 10 different data sizes n as:

Case: xxx Size:yyy Elapsed Time: zzz where, xxx denotes “best”, “worst” or “average”, yyy denotes the data size, and zzz denotes the time (see the answer sheet). You should print 30 statements in total.

Notes

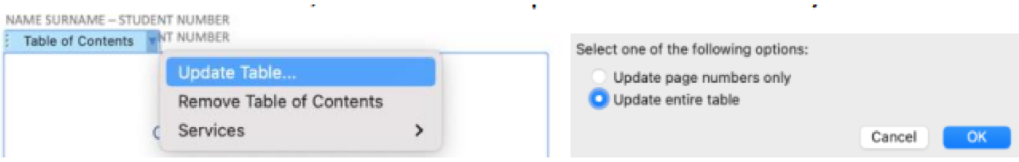
- This Project is designed to be completed by groups of two students

Deadline

- 01.12.2022 23:59. Deadline is strict.
- No late submission will be accepted.

Submission

- The answers of all the questions must be collected into the answer sheet that is provided to you. Please follow the headings, and type your writings under the appropriate heading.
- Please update the table of contents after your report is ready (Click on the arrow on the top of the contents table, then select Update Entire Table).



- Prepare the answers using a word processor; not in handwritten.
- Both students in the group MUST submit the project. The submissions of the two students (i.e. the code documents and the answer documents submitted by the two students) must be exactly the same. A submission by a student implies that she/he has contributed to nearly half of the submitted codes and answers.
- In addition to the project files, each student in a group must submit a document stating explicitly the parts of the project she/he has worked on. The document must begin with the line "I worked on the following parts in this project:" and all the parts the student has worked on must be explained very clearly and in not less than 10 lines. Do not write general comments such as "we worked on the project together", and the like. Write what you have done in the project explicitly. If both students write the same or similar explanations, their projects will not be graded.
- You should also submit the program code.
- Upload a single zip file on Moodle, which consists of the pdf file of the answers, a single file that is the program code and the document that explains the parts of the project you worked on. - Name the files as follows:

  o Answers → NameSurname.pdf (e.g. TungaGungor.pdf)
  o Program code → NameSurname.xxx (xxx: depending on the language)
  o Document → NameSurnameMyWork.pdf
  o Zip file to upload → NameSurname.zip (Zip file that includes the above three files.)

- Each group must answer the exam themselves, without any interactions with others. No materials from resources (internet, books, etc.) are allowed to be used.