CMPE 300 PROJECT 1

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

```
function Example (X[0:n-1])
Input : X[0 : n−1] (a list of size n)
Output : y (an integer)
   y ← 0
   for i ← 0 to n − 1 do
      if X[i] = 0 then                                  (1)
         for j ← i to n − 1 do                          (2)
            for k ← n downto 1 by k ← ⌊k/2⌋ do
               y ← y+1                                  (3) (4)
            endfor
         endfor
      else
         for m ← i to n-1 do                            (2)
            for t ← 1 to n do
               x ← n
               while x > 0 do
                  x ← x-t
                  y ← y+1                               (4)
               endwhile
            endfor
         endfor
      endif
   endfor
   return (y)
end Example
```

Theoretical analysis

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

i) Basic operation is the comparison marked as (1)
ii) Basic operations are the two loop incrementations marked as (2)
iii) Basic operation is the assignment marked as (3)
iv) Basic operations are the two assignments marked as (4)

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 12 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, 10, 50, 100, 200, 300, 400, 500, 600, 700}, for three types of inputs (best-case input, worst-case input, average-case input). For each, record the actual execution time (in miliseconds, 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 average-case input, you can generate a random array X[0:n-1] formed of 0s and 1s according to the probability distribution explained here. 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 12 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 Elapsted 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.

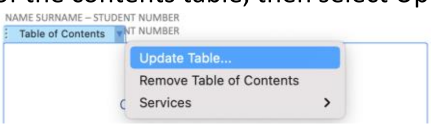
Notes

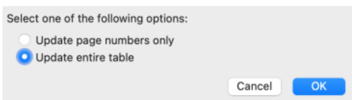
- This Project is designed to be completed by pair of two students.

Deadline

- 16.11.2021 Tuesday 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.
- Submissions from STUDENT 1 is enough. (Check this link.)
- You should also submit the program code.
- Upload a single zip file on Moodle, which consists of the pdf file of the answers and a single file that is the program code.
- Name the files as follows:
  - Answers → NameSurname.pdf (e.g. TungaGungor.pdf)
  - Question 1 code → NameSurname.xxx (xxx: depending on the language)
  - Zip file to upload → NameSurname.zip
- **Each group must answer the exam themselves, without any interactions with others. No materials from resources (internet, books, etc.) are allowed to be used.**
