CSCI 3104 Spring 2022
Instructors: Profs. Chen and Layer

Quiz 12 - Nested Independent Loops

Due Date .... March 4
Name .... **Your Name**
Student ID .... **Your Student ID**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students**. **Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.

2 Standard 12 - Nested Independent Loops

Problem 1. Analyze the runtime of the following algorithm. Clearly derive the runtime complexity function $T(n)$ for this algorithm, and then find a tight asymptotic bound for $T(n)$ (that is, find a function $f(n)$ such that $T(n) \in \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops. [**Note:** $A[n, n]$ denotes an n-by-n matrix with entries $a(i, j),\ 1 \le i \le n,\ 1 \le j \le n$.]

```
1: column_power_sum(A[n, n])
2:   sum = 0
3:   for i = 1, i <= n
4:       i = i + 1
5:       for j = 1, j <= n
6:           j = 2 * j
7:           sum = sum + a(i, j)
8:   return sum
```

Answer. At line 5 the inner printing j loop takes 1 step to initialize
This loop terminates when $2^k > n$. For solving for k we obtain that k¿$\log_2 n$. As the loop takes at least one iteration to compare j to n, we have to have the loop taking $(k > \log_2 n)$+1 iterations. In each iteration the following happens

- In each iteration the comparison j$\le$ n takes one step.
- The update of j=2*j takes two steps, one for the math and one for the new assignment of j
- The update of sum takes three steps one for the math one calling the indexes of a and one for the new assignment of j

So the run time complexity of the inner j-loop is $1 + \sum_{j=1}^{(\log_2 n)+1} 1 + 2 + 3 = \sum_{j=1}^{(\log_2 n)+1} 6 = 1 + 6((\log_2 n) + 1) = 5 + 6(\log_2 n)$

We now analyze the outer loop

- Initializing the outer loop takes 1 step
- The outer loop runs for n iterations. At each iteration the loop does the following
- The comparison $i \le n$ takes 1 step
- The update $i = i + 1$ takes 2 steps: one step to evaluate $i + 1$ and one step for the assignment
- The body the i-loop consists solely of the j loop

So the run time complexity function is

$$\begin{aligned} T(n) &= 1 + \sum_{i=1}^{(\log_2 n)+1} (1 + 2 + 5 + 6(\log_2 n)) \\ &= 1 + \sum_{i=1}^{(\log_2 n)+1} (8 + 6(\log_2 n) \\ &= 1 + \sum_{i=1}^{(\log_2 n)+1} 8 + \sum_{i=1}^{(\log_2 n)+1} 6((\log_2 n) + 1) \\ &= 1 + 8((\log_2 n) + 1) + 6((\log_2 n) + 1)(\log_2 n) \end{aligned}$$

So $T(n) \in \Theta(n^2)$ □