

Quiz 13 - Nested Dependent Loops

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

Contents

1	Instructions	1
2	Standard 13 - Nested Dependent Loops	2

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 13 - Nested Dependent 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.

```
1: procedure foo(integer n):
2:   for i = 1, i <= 2*n
3:     i = i + 1
4:     for j = 1, j <= i
5:       j = j + 1
6:       print 'foo'
```

Answer. We start by analyzing the inner loop. Assume when analyzing the inner loop that i is fixed. The j loop does the following.

- The initialization of the loop takes 1 step
- Observe that the j -loop takes i iterations. At each iteration the loop does the following
- The comparison $j \leq i$ takes 1 step
- The update $j = j + 1$ takes 2 steps
- the body of the loop consist of a single print statement which takes 1 step

So the run time complexity of the j -loop is $1 + \sum_{j=1}^i (1 + 2 + 1) = 1 + \sum_{j=1}^i 4 = 1 + 4i$

We now analyze the outer loop

- Initializing the loop takes 1 step
- Observe that the i loop takes $2n$ iterations at each iteration the loop does the following
- The comparison $i \leq 2n$ takes 1 step
- The update $i = i + 1$ takes 2 steps
- the body of the loop consists solely of the j -loop

So the run time complexity function $T(n)$ is

$$\begin{aligned} T(n) &= 1 + \sum_{i=1}^{2n} n(1 + 2 + (1 + 4i)) \\ &= 1 + \sum_{i=1}^{2n} n(4 + 4i) \\ &= 1 + \sum_{i=1}^{2n} n4 + 4 \sum_{i=1}^{2n} ni \\ &= 1 + 8n + 8 \frac{n(n+1)}{2} \end{aligned}$$

Thus $T(n) \in \Theta(n^2)$

□