CS3000: Algorithms & Data — Summer 2025 — Laney Strange

Recitation 1
Date: May 6th, 2025

Name: Sample Solution

- Recitation problems are for practice only. We'll go over the solutions during your scheduled recitation on Tuesday!
- We will provide `.tex` starter files for recitations, just as we do for homeworks. For most recitations, we strongly encourage you to work out your solution in LaTeXto practice with typesetting. For Recitation 1, some of the problems are specifically *for* practicing with LaTeXand typesetting is part of the solution!
- Collaboration is strongly encouraged during recitation!

Problem 1. *LaTeX Math*

Read through the CS3000 LaTeX Overview

Use LaTeX to typeset the following math snippets:

- A fraction with $n(n+1)$ in the numerator and 2 in the denominator.

  $\frac{n(n+1)}{2}$

  ```
  $\frac{n(n+1)}{2}$
  ```

- The sum as i goes from 1 to n of i^2.

  Solution:

  $\sum_{i=1}^{n} i^2$

  $$\sum_{i=1}^{n} i^2$$

  ```
  $\sum_{i=1}^{n} i^2$
  \[ \sum_{i=1}^{n} i^2  \]
  ```

- The binomial expansion of $(a+b)^4$

  Solution:

  $(a+b)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$

  ```
  $(a+b)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$
  ```

- The quadratic formula.

  Solution:

  $x = \frac {-b \pm \sqrt{b^2 -4ac}} {2a}$

  ```
  $x = \frac {-b \pm \sqrt{b^2 -4ac}} {2a}$
  ```

Problem 2. *LaTeX Pseudocode*

Read through the CS3000 LaTeX Pseudocode Guide

Complete the pseudocode below for linear search. It should return the position in the array where the key is found, or NIL if it's not found.

LINEARSEARCH(A, n, key)

for $i = 1$ **to** n

Solution:

LINEARSEARCH(A, n, key)

for $i = 1$ **to** n
    if $A[i] == key$
        return i
return NIL

```
\begin{codebox}
\Procname{{$\proc{Search}(A,n,key)$}}
\li \For $i \gets 1 \To n$
\li \Do
\If $A[i] == \id{key}$
\li \Then
    \Return $i$
\End
\End
\li \Return \const{Nil}
\end{codebox}
```

Problem 3. *Wall-Clock Time*

For each function $f(n)$ and time t in the following table, determine the largest input size n of a problem that can be solved in time t, assuming that the algorithm takes $f(n)$ milliseconds (there are 10^3 milliseconds per second).

	1 second	1 minute	1 day
n			
n^2			
2^n			

Solution:

	1 second	1 minute	1 day
n	10^3	$6 \cdot 10^4 = 60000$	$864 \cdot 10^5 = 86400000$
n^2	$\sqrt{10^3} = 31$	$\sqrt{6 \cdot 10^4} = 244$	$\sqrt{864 \cdot 10^5} = 9295$
2^n	$\lg 10^3 = 9$	$\lg(6 \cdot 10^4) = 15$	$\lg(864 \cdot 10^5) = 26$