

**Department of Computing**

**Algorithms and Data Structures**

**(55-508226-AF-20212)**

**Name: R. C. Edwards**

**Student ID: 30012275**

**Table of Contents**

[1. Project 1 4](#_Toc83301441)

[1.1 Algorithms in ADL 4](#_Toc83301442)

[1.2 Software and its presentation, including testing (and video link) 4](#_Toc83301443)

[1.4 Descriptive Report, including artefacts 4](#_Toc83301444)

[1.4.1 Transitioning algorithms to implementation 4](#_Toc83301445)

[1.4.3 Problem-solving strategy 4](#_Toc83301446)

[1.5 Incorporation of formative feedback 4](#_Toc83301447)

[2. Project 2 4](#_Toc83301448)

[2.1 Descriptive report, including artefacts 4](#_Toc83301449)

[2.1.1 Time-Space Trade-off Choices made 4](#_Toc83301450)

[2.1.2 Computational Complexity 4](#_Toc83301451)

[2.2 Incorporation of formative feedback 4](#_Toc83301452)

[3. Project 3 4](#_Toc83301453)

[3.1 Algorithms in ADL 4](#_Toc83301454)

[3.2 Data Structures 4](#_Toc83301455)

[3.3 Software and its Presentation, including testing (and video link) 4](#_Toc83301456)

[3.4 Descriptive Report, including artefacts 4](#_Toc83301457)

[3.4.1 Transitioning algorithms to implementation 4](#_Toc83301458)

[3.4.2 Problem-solving strategy 4](#_Toc83301459)

[3.5 Incorporation of formative feedback 4](#_Toc83301460)

[4. Project 4 4](#_Toc83301461)

[4.1 Proposed Algorithms 4](#_Toc83301462)

[4.1.1 Elevator Algorithm 4](#_Toc83301463)

[4.1.2 The Elevator Algorithm in ADL 4](#_Toc83301464)

[4.2 Software and its Presentation, including testing (and video link) 4](#_Toc83301465)

[4.3 Descriptive Report, including artefacts 4](#_Toc83301466)

[4.3.1 Experiment Strategy 4](#_Toc83301467)

[4.3.2 Experiment Results 4](#_Toc83301468)

[4.3.3 Findings and Discussion 4](#_Toc83301469)

[4.4 Incorporation of formative feedback 5](#_Toc83301470)

[5. Project 5 5](#_Toc83301471)

[5.1 Algorithms 5](#_Toc83301472)

[5.1.1 Introduction to Genetic Algorithm 5](#_Toc83301473)

[5.1.2 Proposed Genetic Algorithm in ADL 5](#_Toc83301474)

[5.2 Software and its Presentation, including testing (and video link) 5](#_Toc83301475)

[5.3 Descriptive Report, including artefacts 5](#_Toc83301476)

[5.3.1 Experiment Strategy 5](#_Toc83301477)

[5.3.2 Experiment Results 5](#_Toc83301478)

[5.3.3 Findings and Discussion 5](#_Toc83301479)

[5.4 Incorporation of formative feedback 5](#_Toc83301480)

# 1. Project 1

## 1.1 Algorithms in ADL

## 1.2 Software and its presentation, including testing (and video link)

## 1.3 Descriptive Report, including artefacts

### 1.4.1 Transitioning algorithms to implementation

### 1.4.3 Problem-solving strategy

## 1.4 Incorporation of formative feedback

# 2. Project 2

## 2.1 Descriptive report, including artefacts

### 2.1.1 Time-Space Trade-off Choices made

### 2.1.2 Computational Complexity

## 2.2 Incorporation of formative feedback

# 3. Project 3

## 3.1 Algorithms in ADL

## 3.2 Data Structures

## 3.3 Software and its Presentation, including testing (and video link)

## 3.4 Descriptive Report, including artefacts

### 3.4.1 Transitioning algorithms to implementation

### 3.4.2 Problem-solving strategy

## 3.5 Incorporation of formative feedback

# 4. Project 4

## 4.1 Proposed Algorithms

### 4.1.1 Elevator Algorithm

### 4.1.2 The Elevator Algorithm in ADL

## 4.2 Software and its Presentation, including testing (and video link)

## 4.3 Descriptive Report, including artefacts

### 4.3.1 Experiment Strategy

### 4.3.2 Experiment Results

### 4.3.3 Findings and Discussion

## 4.4 Incorporation of formative feedback

# 5. Project 5

## 5.1 Algorithms

## 5.1.1 Introduction to Genetic Algorithm

## 5.1.2 Proposed Genetic Algorithm in ADL

## 5.2 Software and its Presentation, including testing (and video link)

## 5.3 Descriptive Report, including artefacts

### 5.3.1 Experiment Strategy

### 5.3.2 Experiment Results

### 5.3.3 Findings and Discussion

## 5.4 Incorporation of formative feedback

**Project 1: Pascal’s Triangle**

* 1. Algorithms in ADL

* Solution 1:

**procedure** displayTriangle(**declare** rowNumber)

**for** index1 0 **to** rowNumber **by** 1 **do**

**declare** numberToPrint 1

**for** index2 ← rowNumber - index1 **to** 1 **by** -1 **do**

**print** (“ “);

**end**

**for** index2 ← 1 **to** index1 **by** 1 **do**

**print** (numberToPrint);

numberToPrint = numberToPrint \* (i-j)/(j+1);

**end**

**print** (“ “);

**end**

* 1. Software and its Presentation, including Testing (and video link)

Video link and test code goes here

* 1. Descriptive Report, including Artefacts

1.31 Transitioning algorithms to implementation

1.32 Problem-solving strategy

1. Get number of rows the user wants to print.

2. Begin a for loop (with i initialised to zero) that runs as many times as there are rows, so that we can print the contents of the triangle.

3. Initialise the first number of the row, which would be 1 as that is what each row begins with.

4. Add a for loop that shifts the first number to the left depending on what number row it is. The higher the number, the less it is shifted. This is done to format the triangle like a pyramid.

5. Begin another for loop with variable j (initialised to zero) that runs until it is equal to i in the parent loop.

6. Print the number, with a space between it and the next number.

7. Calculate the binomial coefficient of the position of the next number. For example:

* If we were calculating the middle number of the fifth row, we would use the previous number (4) and the iteration numbers of the for loop that we are in, and the for loop that we are nested in (i and j). Our formula here would be 4 \* (i - j) / (j + 1). To get to this middle number, we would be on our fifth iteration of i and our second iteration of j, which would make i equal to 4 and j equal to 1. Subtracting j from i, then dividing the result of that by the value of j + 1 gives us the number we need to multiply 4 by - which we can then print after 4.
* 4 \* (4 - 1) / (1 + 1)

= 3 = 2

4 \* (3 / 2)

= 1.5

4 \* 1.5

= 6, which is what we will print next

1. Once the nested loop is finished, print a new line to keep the triangle formatted correctly, and move onto the next iteration until all of the rows have been printed.

**Project 2: Algorithmic Complexity and Space-Speed Trade Off**

2.1 Descriptive Report, including Artefacts

2.1.1 Computational Complexity

**for** i **←** 1 **to** n **by** 1 **do this line is O(n)**

**for** j **←** 1 **to** i **by** 1 **do this line is O(n)**

**for** k **←** 1 **to** j **by** **this line is O(n)**

x = x + 1 **this line is O(1)**

**end**

**end**

**end**

This algorithm contains 3 nested for loops, each loop being N. After an iteration of all three loops, x is incremented.

I to n, j to I, k to j

Make it a graph.

2.1.2 Time-Space Trade-Off Choices Made