# Data Structure Assignment 7

|  |  |  |
| --- | --- | --- |
| **ID:** E14066282 | **Name:** 溫梓傑 | **Department:** ME 110 |

## Result Screenshots

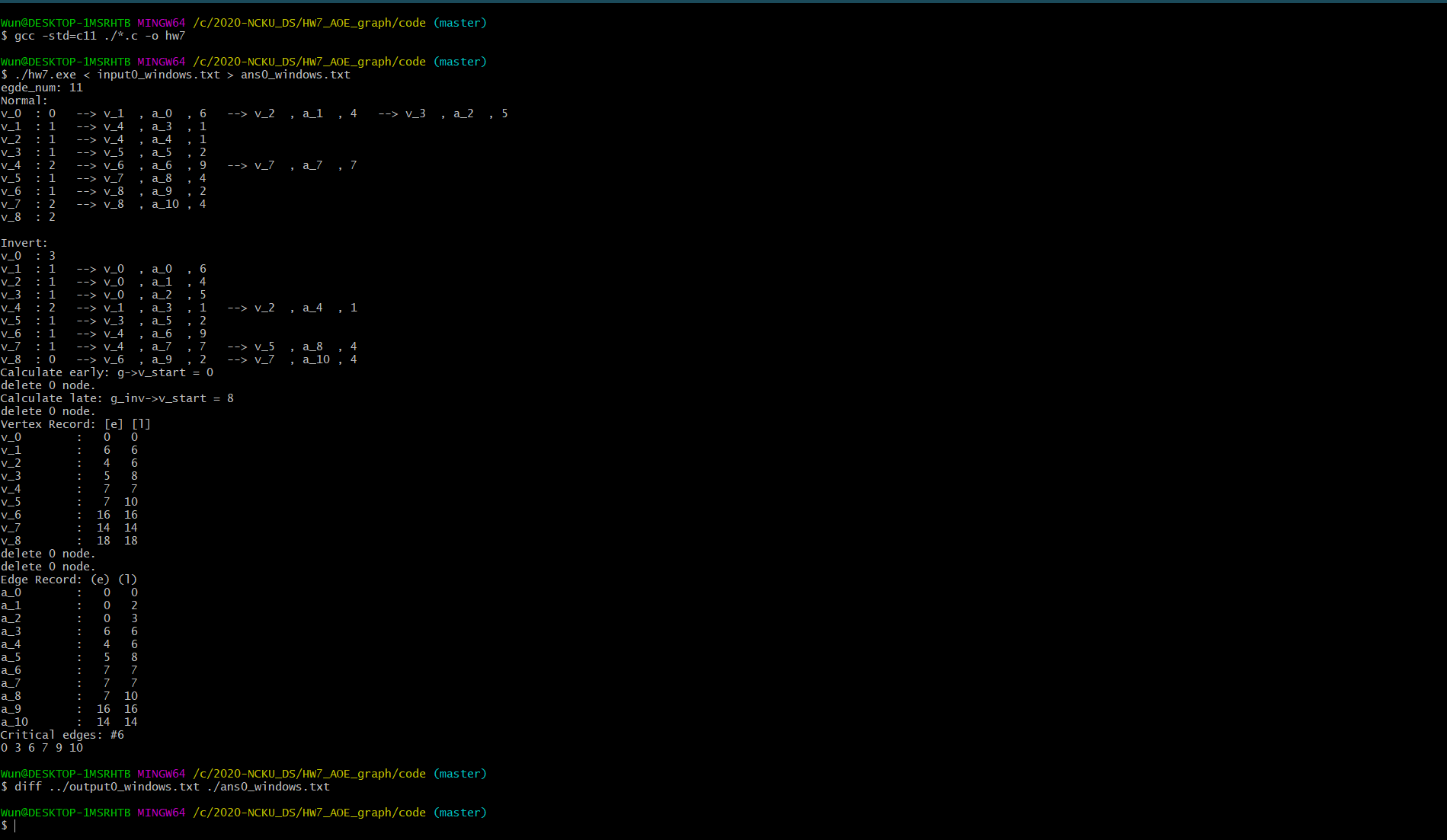


Figure Screenshot of command line

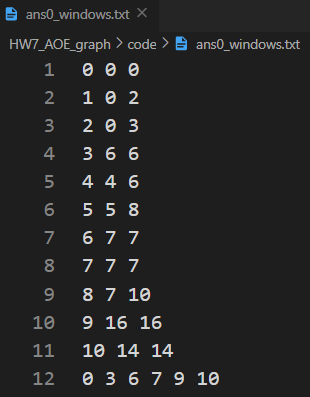


Figure ans\_output0\_windows.txt

## Program Architecture

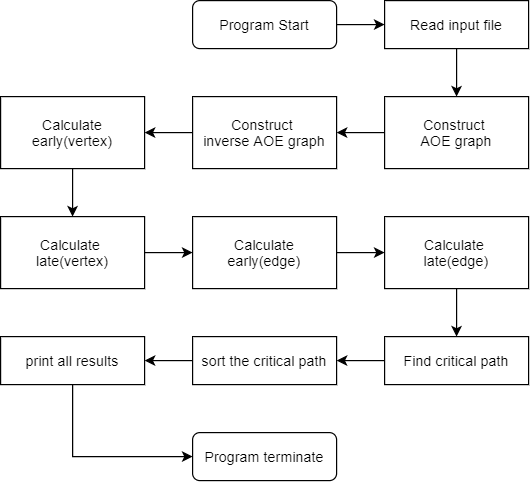


Figure Flow chart of hw7

## Program Functions

📒 graph.h

### 📁Data Structure of graph, vheadnode, gnode

**typedef** **struct** **gnode**

{

**int** vertex;

**int** weight;

**int** edge\_code; // A -> B: B.edge\_code = E(A,B)

**struct** **gnode** \*link;

} **gnode**;

**typedef** **struct** **vheadnode**

{

**int** count;

**gnode** \*link;

} **vheadnode**;

**typedef** **struct** **graph**

{

**vheadnode** \*vhead;

**int** v\_num;

**int** v\_start; // default = 0

} **graph**;

**graph** \*create\_aoe(**int** v\_num)

Construct a graph object.

### 📐Parameters

v\_num

The size of vheadnode array.

### ↩Return Value

Returns the new pointer the graph.

* If construction fails, returns NULL.

**weight, edge\_code**

**gnode** \*create\_gnode(**int** v\_end, **int** weight, **int** edge\_code)

A gnode represents a weighted-edge and the succeeding vertex.

Construct a gnode object.

### 📐Parameters

v\_end

The vertex code which is the end of edge.

weight

The weight of the edge.

edge\_code

The code of the edge which is corresponding to the input.

### ↩Return Value

Return the new pointer of gnode.

**int** find\_vstart(**graph** \*g)

Find the start vertex of the graph.

### 📐Parameters

g

The graph to be found.

### ↩Return Value

The start code of the vertex of the graph.

**graph** \*construct\_aoe(**int** \*\*g\_in, **int** e\_num)

Construct an AOE graph object.

### 📐Parameters

g\_in

The description about the graph. (e\_code, v\_start, v\_end, weight)

e\_num

The number of the edge.

### ↩Return Value

Return the pointer of the AOE graph.

**graph** \*construct\_aoe\_inv(**int** \*\*g\_in, **int** e\_num)

Construct an inverse AOE graph object.

### 📐Parameters

g\_in

The description of edge about the graph. (e\_code, v\_start, v\_end, weight)

e\_num

The number of the edge.

### ↩Return Value

Return the pointer of the inverse AOE graph.

**void** cal\_edge\_rec(**int** \*e\_rec, **int** v\_rec[100], **graph** \*g, **int** mode)

Calculate the early edge or the late edge.

### 📐Parameters

e\_rec

The record of the edges.

v\_rec

The record of the vertex.

graph

The AOE graph to be analyzed.

mode

0 represent the calculation of early.

1 represent the calculation of late.

### ↩Return Value

None.

## Program Design

本次作業使用Adjacency lists來儲存圖形，其資料結構請參考上面敘述，基本上這次作業完全依照講義上的方法來實現，使用Topological order來計算所有vertex的early與late。

而在計算所有edge的early與late時，使用廣度優先的造訪方法來依序存取edge與vertex，再套用課本公式：

*early(i) = earliest[k]*

*late(i) = latest[l] - duration of activity ai*

## Operating System

Windows 10

## Compiler

gcc.exe (MinGW.org GCC Build-20200227-1) 9.2.0

## Compile

gcc -std=c11 ./\*.c -o hw7

## Run

./hw7.exe < input.txt > output.txt