Data Structure

Homework 12

40671215H 呂沛融

書面報告:

這次程式作業是用 adjacency list完成，並且利用Min-Priority Queue 輔助prim’s演算法。

procedure功能：

此程式中，有以下procedures

(分為兩部分，一為binary Queue，一為prim’s演算法)

binary Queue-

void swap(struct HeapNode &p1, struct HeapNode &p2);

//交換

int FindPosition(int node); //找到位置

int GetParentNode(int node); //得到parent

void MinHeapify(int node, int length);

//利用min heap調整tree

void BuildMinHeap(std::vector<int> array);

//每一個有child的點都進行過一次MinHeapify()

void DecreaseKey(int node, int newKey);

//調整Min Heap tree中的點之Key值

int ExtractMin(); //回傳最小key的點index再移除

int Minimum() //得到最小值

bool IsHeapEmpty() //確認heap存在

prim’s演算法-

void InitializeSingleSource(int Start);

//起始點

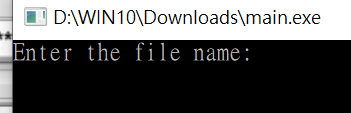
void AddEdge(int from, int to, int weight);

//建立adjacency list

void Prim\_MinQueue(int Start); //演算法過程

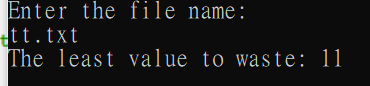
輸出入介面說明：

1. 當顯示以下介面時，請輸入要讀的檔案：



若沒有讀到檔案會顯示錯誤，重複輸入直到讀對檔。

1. 在讀到檔案情況下，程式會輸出最少的總消耗能量。



※若輸入的編號型態錯誤會中止程式。

程式時間複雜度分析：

將graph用adjecency list表示後，可以使用min heap去做中間取最小邊的動作，利用heap取最小值的時間複雜度為O(logn) (想像為樹高)，因此在prim’s演算法中(設好graph的情況下)只利用了O(logn)去生成MST(取邊的最小值)，演算法的時間複雜度可視為O(logn)