**Programming Assignment#3 Report**

**NTUEE**

**B04501095**

**黃平瑋**

1. **演算法流程 (Algorithm Flow)**

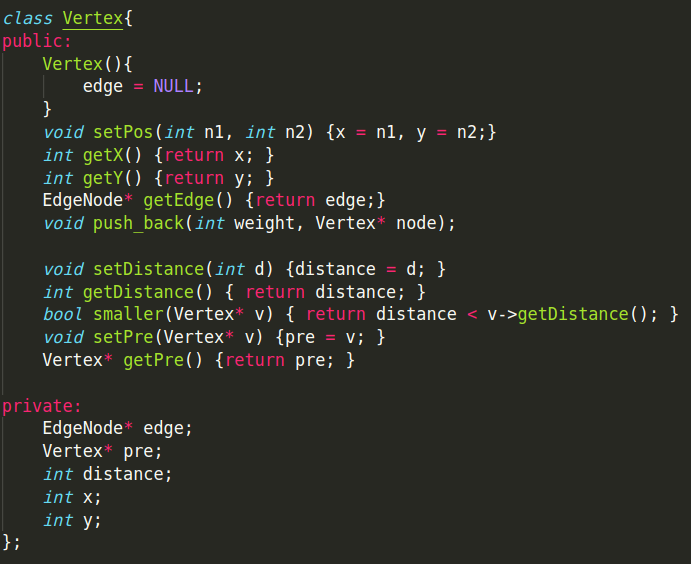
在這次的作業的繞線問題，我將他轉換成graph algorithm中**shortest path** **problem**，同時將繞線的格子轉換成一個graph，每個格子代表一個vertex，格子與格子間的邊界為一個edge，edge的權重可用不同的cost function表示，常見的有線性或是指數關係。

針對輸入的繞線點，算出各繞線起始到結束的距離，取距離的方式為|x座標差|+|Y座標差|，並由小到大排列，由最小的開始跑**Diijkstra's algorithm**，在從結束點的座標開始，利用predecessor backtrace，直到回到起始點為止，即可找到兩點間的繞線。在更新edge權重完畢後，再換到下一條繞線，直到所有線皆繞完為止。

1. **資料結構 (Data Structure)**

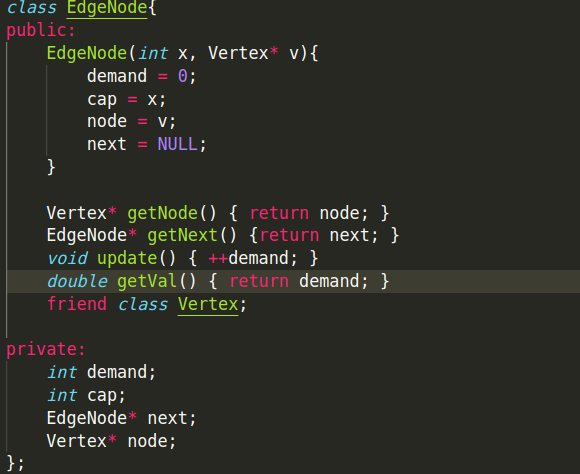
**(a). Vertex**

針對這個問題我將他model成一個graph中的shortest path 問題，grid中的每個格子轉換成一個vertex，採用adjacent list實做weighted edge，以下是他的資料結構：



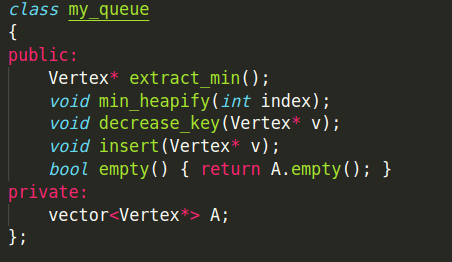
**(b). edge**

另外針對weighted edge, 實做的一個class，**demand** 紀錄通過的繞線數量，因為是linked list的結構，所以需要EdgeNode\* **next**指向下一個edge，而Vertex\* node 則只向這個edge所指向的node



**(c). priority queue**

在執行dijkstra's algorithm時，需要用到priority queue，在此次作業我使用的是binary heap,並實做 EXTRACT\_MIN, DECREASE\_KEY, INSERT等功能



1. **問題與討論 (Discussion)**

針對程式執行的結果，以不同的edge cost function比較，

cost function A: cost = 2 ^ (demand / capacity)

cost function B: cost = demand / capacity

在實驗後發現指數的cost function不論在overflow, wirelength表現皆比較好，下表是實驗的結果。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| grid | #net | runtime | cost function A | | cost function B | |
| overflow | wirelength | overflow | wirelength |
| 4x4 | 3 | 0 | 0 | 13 | 0 | 15 |
| 5x5 | 10 | 0 | 0 | 36 | 0 | 40 |
| 10x10 | 40 | 0 | 0 | 244 | 0 | 304 |
| 20x20 | 1500 | 0.24 | 0 | 19872 | 0 | 30334 |
| 60x60 | 10500 | 87.42 | 50920 | 314878 | 53561 | 320022 |