

競賽題庫 - d908 最佳路徑

<div><div>ZERO JUDGE</div><div>分類題庫 解題動態 排行榜 討論區 競賽區</div><div>搜尋題目關鍵字、題號...</div><div>411440521TKUEE 看來過倒你叫誰 0</div></div>						
10881169	joshlee92102... (411440521TKUEE 看...)	d908. 4. 最佳路徑 - 99學年度北基區資訊學科 能力競賽	AC (3ms, 380KB)	CPP	2022-12-19 15:53	
10881166	joshlee92102... (411440521TKUEE 看...)	d908. 4. 最佳路徑 - 99學年度北基區資訊學科 能力競賽	NA (score:80%)	CPP	2022-12-19 15:52	
10881162	joshlee92102... (411440521TKUEE 看...)	d908. 4. 最佳路徑 - 99學年度北基區資訊學科 能力競賽	NA (score:80%)	CPP	2022-12-19 15:48	
10881148	joshlee92102... (411440521TKUEE 看...)	d908. 4. 最佳路徑 - 99學年度北基區資訊學科 能力競賽	NA (score:80%)	CPP	2022-12-19 15:45	
10881127	joshlee92102... (411440521TKUEE 看...)	d908. 4. 最佳路徑 - 99學年度北基區資訊學科 能力競賽	NA (score:20%)	CPP	2022-12-19 15:43	

這題其實是常見的最小路徑，不過反過來變成最長而已。
先來整理一下常見解最短路徑的方法大概有哪些

最短路徑常用方法

方法	Floyd-Warshall's Algorithm	Dijkstra's Algorithm	Bellman-Ford's Algorithm
用途	多源最短路徑	單源最短路徑	單源最短路徑
實作方法	枚舉+dp	貪婪法	dp
參考連結	https://ithelp.ithome.com.tw/articles/10209186	https://ithelp.ithome.com.tw/articles/10209593	https://ithelp.ithome.com.tw/articles/10209748

單源最短路徑 (確定起點的最短路徑)

邊權為正時適合使用 Dijkstra's Algorithm

邊權為負時適合使用 Bellman-ford's Algorithm

多源最短路徑 (不指定起點的最短路徑)

適合使用 Floyd-Warshall's Algorithm

因為是求多個點與點間的最「長」路徑，所以選 Floyd-Warshall 解會好一點

先直接做一個 Floyd Warshall

```

int Floyd_Warshall() {
    for(int k=0;k<n;k++){
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                // 如果兩數都不為 0，且兩點相加大於原本紀錄長度
                if( (d[i][k]&& d[k][j]) && (d[i][j] < d[i][k]+d[k][j]) )
                    // 則 i 到 j 改為相加(較長)的長度
                    d[i][j] = d[i][k]+d[k][j];
            }
        }
        // 找起點到終點的最長距離
        int M=0;
        for(int i=0;i<n;i++) M=max(M, d[start-'A'][i]);
        return M;
    }
}

```

主程式部分(輸出入)

```

int main() {
    while(cin>>start>>n){
        // 這句在競程比較常見，用來初始化陣列
        memset(d, 0, sizeof(d));
        for(int i=0;i<n;i++){
            cin>>a>>b>>c;
            // 如果測資重複，則取大的存入
            if(c > d[a-'A'][b-'A']) d[a-'A'][b-'A']=c;
        }
        cout<<Floyd_Warshall()<<"\n";
    }
}

```

心得：

這次作業我認為我在實作資料結構和演算法的能力上有了不錯的進步幅度。從開學還只是高中培訓時有記下幾個演算法，但礙於平日補習沒時間好好練習。到現在能夠實作，而且能夠有理由的告訴自己挪出一長段時間好好寫程式，把基礎的知識鞏固好。