

Graph Theory

圖論

圖論

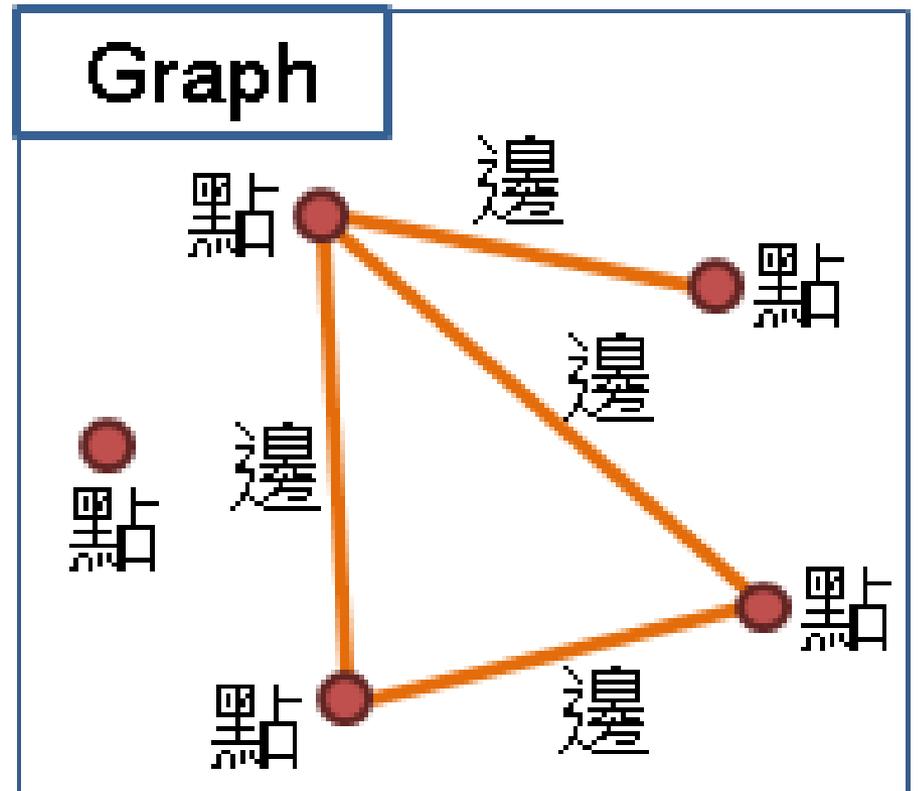
圖論 – 研究圖的理論

圖是甚麼?

圖論

一張圖是由點和邊所構成

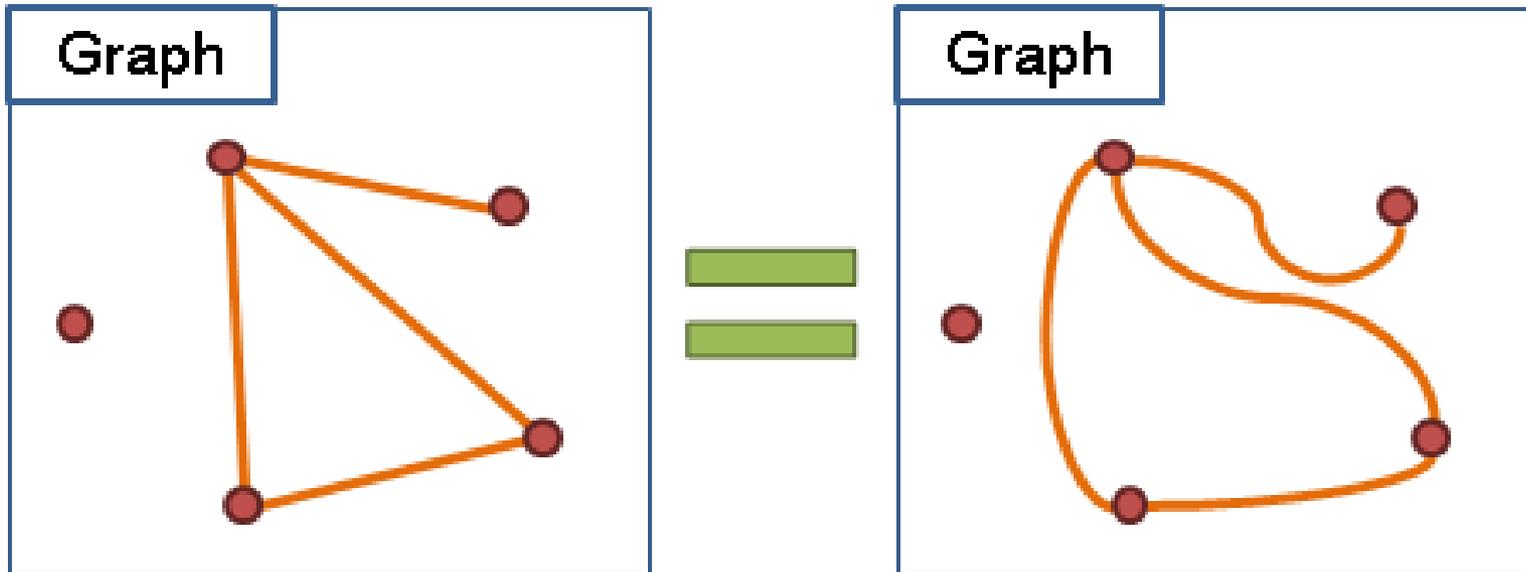
每個點可以當成一個物品



點和點有邊，代表他們存在某種關係

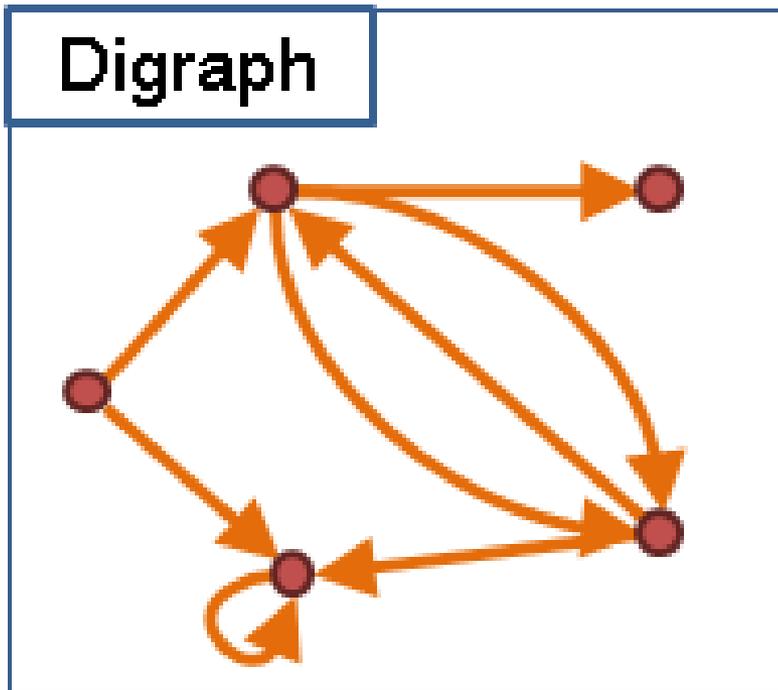
圖論

圖是一個很抽象的概念
只要點和點的關係不變
你可以隨意的畫他



圖論

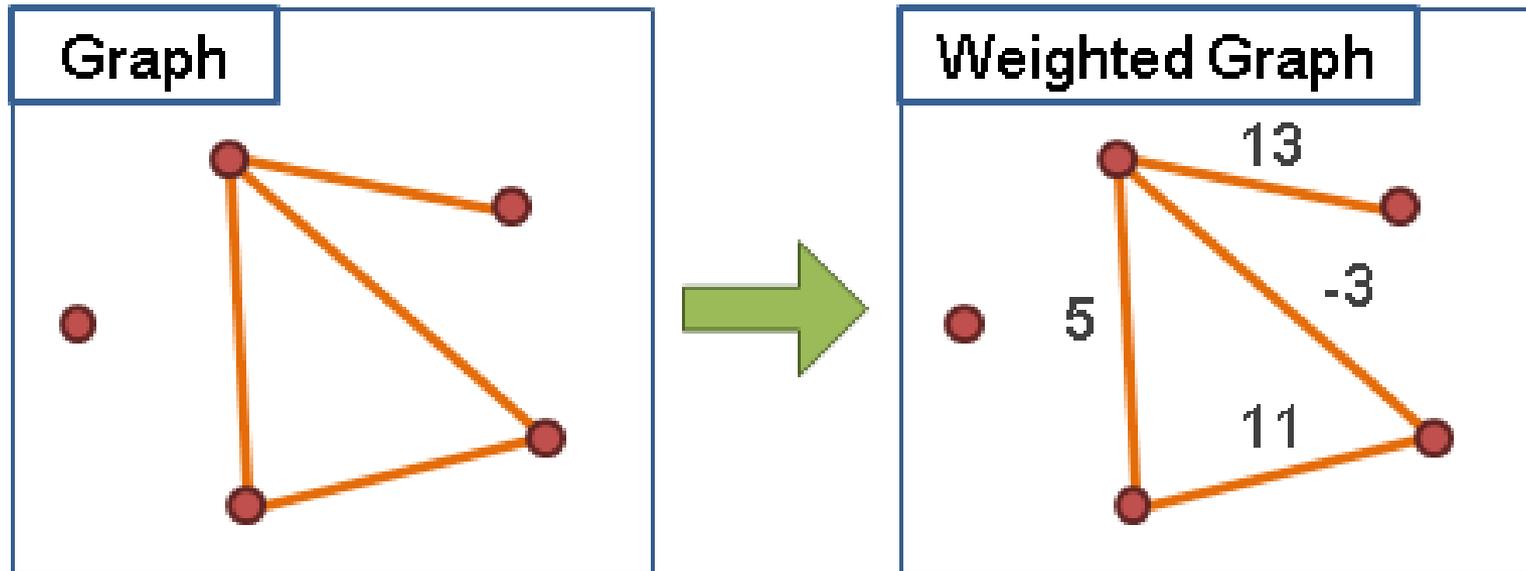
有時候邊會帶有一些特殊的屬性，比如方向性



圖論

有時候邊會帶有一些特殊的屬性，比如長度(成本)

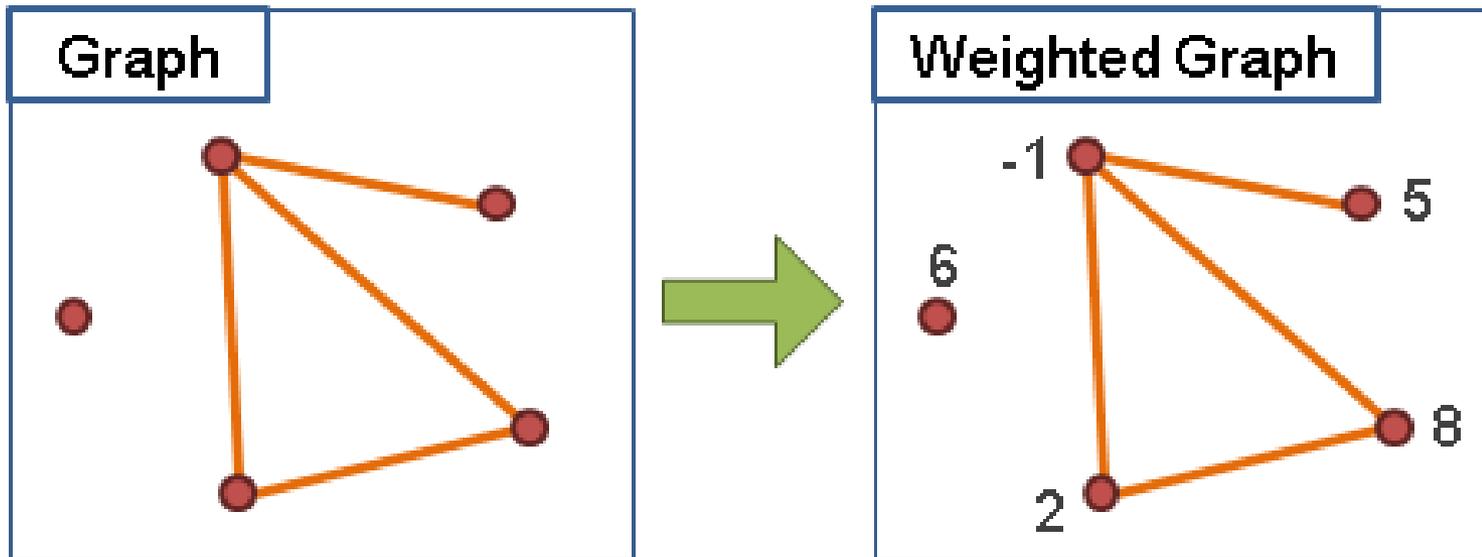
Ex. 城市和城市之間的距離(通行成本)



圖論

有時候點會帶有一些特殊的屬性，比如權重

Ex. 每個關口的手續費

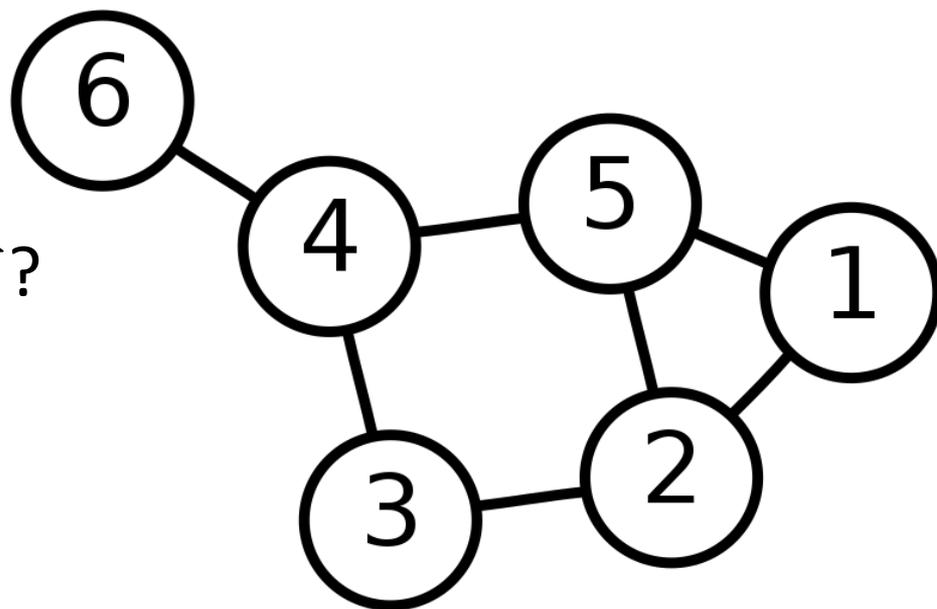


圖論

將複雜的現實世界簡化成圖這個模型
可以很方便做到定性和定量

Ex. 距離問題

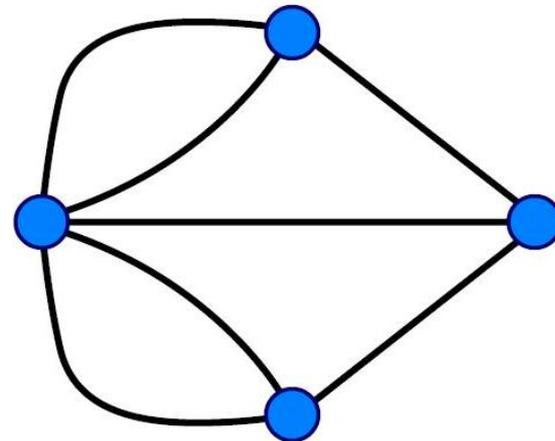
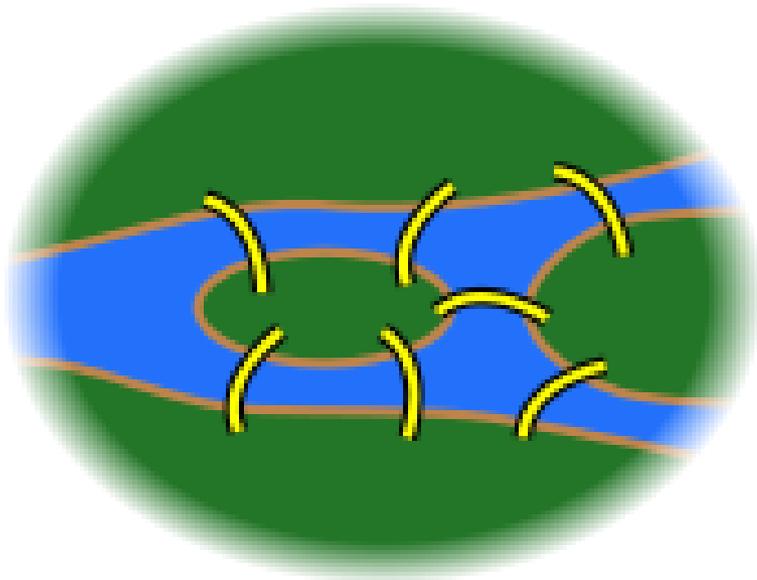
從城市 1 走到城市 6
至少要經過多少個城市？



圖論

將複雜的現實世界簡化成圖這個模型
可以很方便做到定性和定量

Ex. 七橋問題: 能否每座橋只走一次，走完全部的橋?



在程式中儲存一張圖

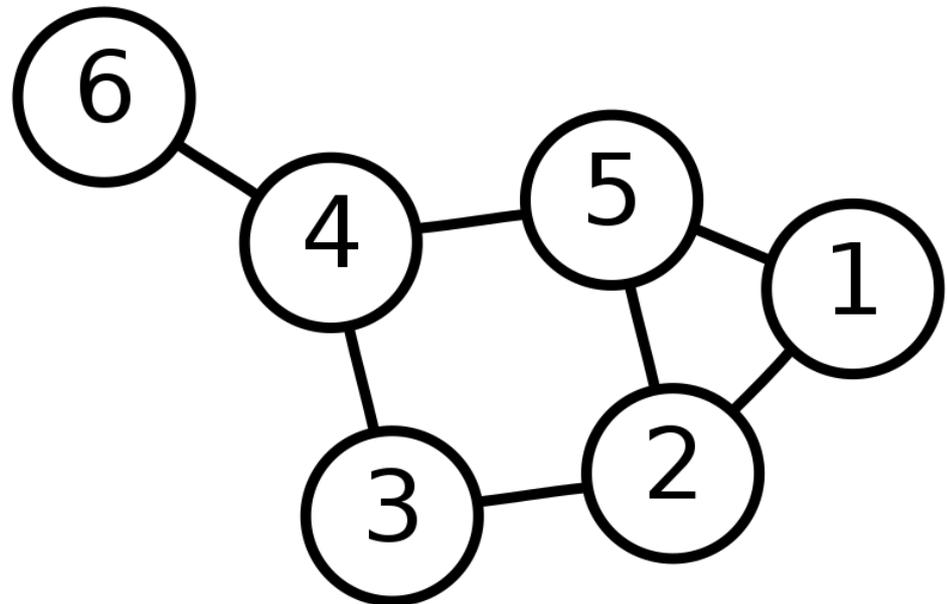
圖的遍歷(Traverse)

BFS, DFS

Breadth-First Search(BFS)

不斷找出尚未遍歷的點當作起點，進行下述行為：

- 一、把起點塞入queue。
- 二、重複下述兩步驟，直到queue裡面沒有東西為止：
 - 甲、queue彈出一點。
 - 乙、找出跟此點相鄰的點，並且尚未遍歷的點，通通（依照編號順序）塞入queue。



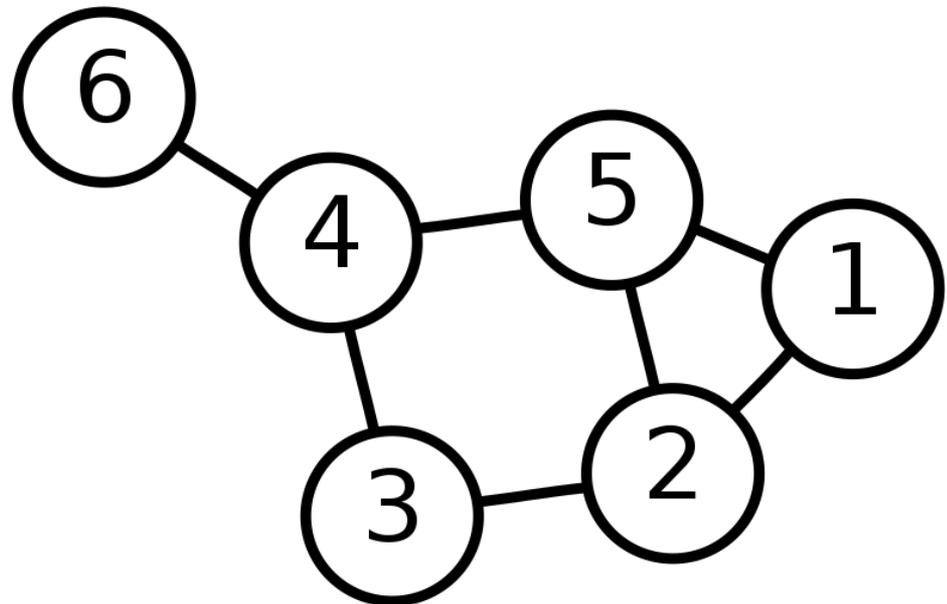
Depth-First Search(DFS)

從某個還沒有被找到的點開始找，用一個遞迴函式來找下去。

DFS(某個點):

枚舉每個相鄰的點:

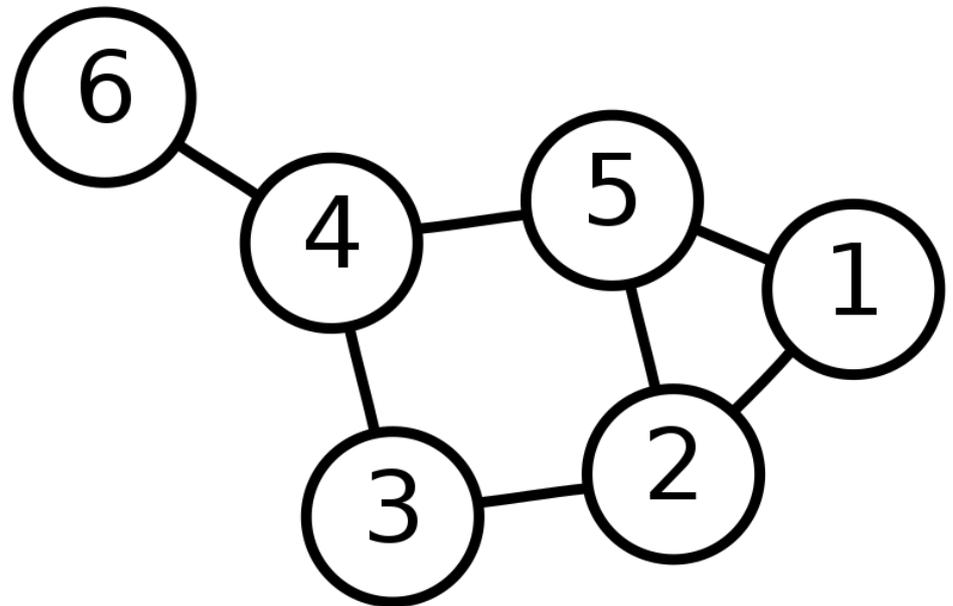
如果還沒有走過，則 DFS(該點)



Implement In C++

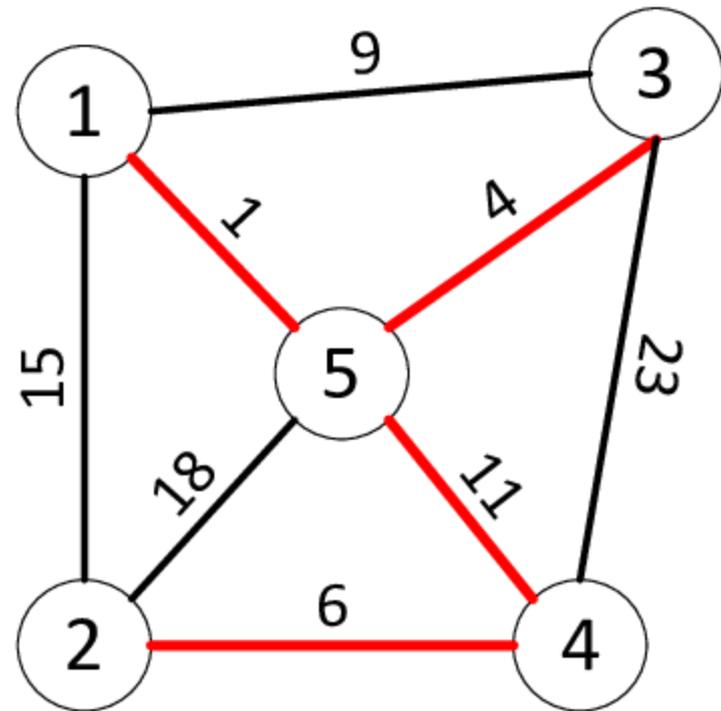
Shortest path with BFS

- **Ex.** 路由: 你要上網，資料要走最近的距離到目標主機，假設每條網路線的延遲相同，要怎麼做？



Shortest path

- **Ex.** 路由: 你要上網，資料要走最近的距離到目標主機，但每條網路線的延遲和頻寬不可能都相同，那要怎麼做？



Priority Queue

- 一樣大家在排隊，但某些人可以插隊
- 可以想成它會自動 `sort` 的 `queue`
- 功能：
 - 插入一個數
 - 查詢最小(或最大)的數
 - 刪除最小(或最大)的數

Shortest path with Dijkstra

- **Ex.** 路由: 你要上網，資料要走最近的距離到目標主機，但每條網路線的延遲和頻寬不可能都相同，那要怎麼做?
- **BFS** 沒有考慮到點和點的距離不同
 - 用一個 **priority queue** 來取代 **queue**
 - 讓距離越近的越優先往下找