

Dijkstra's shortest path algorithm: (只能用在非負權圖)

(固定一點，而該點到所有點的最短距離)

定義：最短距離函數

假設 $G=(V, E)$ 為一個加權圖

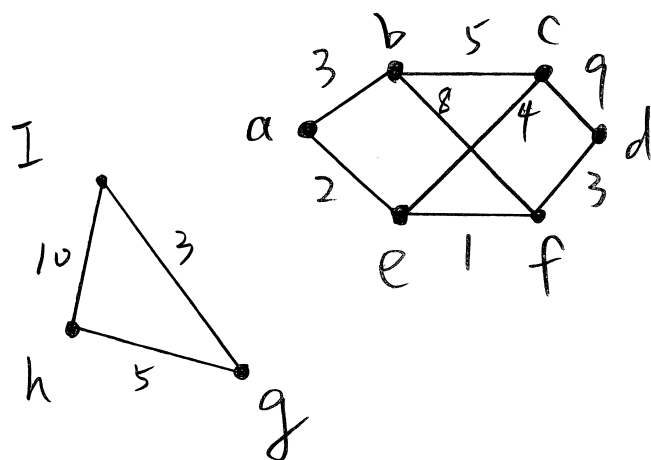
1. $\forall x, y \in V$, $P = (x, v_1, v_2, \dots, v_n, y)$ 為一條從 x 到 y 的路徑，定義 P 為一個長度 (length)，為 $w_t(x, v_1) + w_t(v_1, v_2) + w_t(v_2, v_3) + \dots + w_t(v_{n-1}, v_n) + w_t(v_n, y)$ 。

2. 定義最短距離函數 (the shortest distance function) d ,
 $d: V \times V \rightarrow \mathbb{R}^+ \cup \{0, \infty\}$

解釋：1. $V \times V$ 指 2 個點

2. $\mathbb{R}^+ \cup \{0, \infty\}$ 指正實數且包含 0 和無限大。

例題



$$d(a, a) = 0$$

$$d(a, g) = \infty$$

$$d(a, b) = 3$$

$$d(a, c) = 6$$

$$d(h, I) = 8$$

Pseudo code:

Algorithm: Dijkstra's algorithm

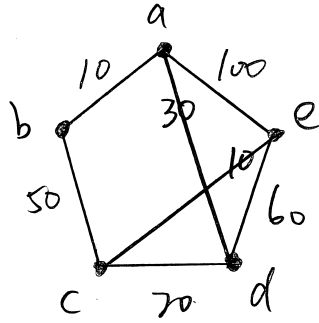
Input: A connected, weighted graph $G = (V, E)$ in which all weights are positive with source vertex a .

Output: The distance $D[v]$ from a to v .

1. $S := \{a\}$
2. $D[a] = 0$
3. for each $v \in V - \{a\}$ do
4. $D[v] := wt(a, v)$ // 指 a 到 v 的一個邊的距離,
 // 不相鄰為 ∞ .
5. while $|S| \neq |V|$ do
 /* Choose a vertex w in $V - S$ such that $D[w]$ is
 minimum, then add w to S . If not, skip it */
6. for each $v \in V - S$ do
7. $D[v] := \min(D[v], D[w] + wt(w, v))$

例題：

利用 Dijkstra's algorithm 求 a 到各點的最短距離：



Ans.

利用 Dynamic programming 方式 表格解，距離右下解標示上個點。
(由左至右，由上而下填表)

$s \backslash D$	b	c	d	e
$\{a\}$	$\textcircled{10_a}$	∞_a	30_a	100_a
$\{a, b\}$	X	60_b	$\textcircled{30_a}$	100_a
$\{a, b, d\}$	X	$\textcircled{50_d}$	X	90_d
$\{a, b, d, c\}$	X	X	X	$\textcircled{60_c}$

$$d(a, b) = 10$$

$$d(a, d) = 30$$

$$d(a, c) = 50$$

$$d(a, e) = 60$$

得出最小生成樹為：

