Greedy Algorithms

Knapsack Problem

มีของ n ชิ้น : $1, 2, 3, \ldots, n$

แต่ละชิ้นมีน้ำหนัก : $w_1, w_2, w_3, \ldots, w_n$

" —— " มีมูลค่า : $v_1, v_2, v_3, \ldots, v_n$

มีถุงเป้ 1 ใบ รับน้ำหนักได้ W

จะเลือกของชิ้นใดดี ใส่ถุง โดยที่

- ถุงไม่ขาด
- ได้มูลค่ารวมสูงสุด.

ให้ ผลเฉลย คือ $x_1, x_2, x_3, \ldots, x_n$

$$0 \leq x_i \leq 1 \qquad i = 1, 2, 3, \ldots, n$$

หา $x_1, x_2, x_3, \ldots, x_n$ ที่

$$\text{maximize} \quad \sum_{i=1}^{n} x_i v_i$$

subject to

$$\sum_{i=1}^{n} x_i w_i \leq W$$

$$0 \leq x_i \leq 1 \qquad i = 1, 2, 3, \ldots, n$$

ตย.

	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
ดีสุดคือ x_i	1.0	1.0	1.0	0	0.8

$W = 100$

$\Downarrow$

$$\sum x_i v_i = 164$$

$$\sum x_i w_i = 100 = W$$

โครงของอัลกอริทึมเชิงละโมบ

```
Greedy ( C ) {
    S ← φ
    while( C ≠ φ AND ! solution(S) ) {
        x ← select(C) **
        C ← C - {x}
        if ( feasible( S ∪ {x}) )
            S ← S ∪ {x}
    }
    if solution(S) then return S
                   else "no solution"
}
```

Knapsack Problem

Greedy strategy :

- โลภมาก - คว้าของ เรียงตามมูลค่าของสูงสุด
- ห่วงถุงขาด - " ———— " น้ำหนัก
  (อยากได้ของหลายๆ ชิ้น)

	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

$W = 100$

① คว้าของแพงก่อน : $\sum v_i x_i = 66 + 60 + 20 = 146$

② " — " เบา " : $\sum v_i x_i = 20 + 30 + 66 + 40 = 156$

③ " — " มูลค่า/น้ำหนัก มากสุดก่อน $\sum x_i v_i$

: $\sum v_i x_i = 66 + 20 + 30 + 48 = 164$

③ ดีกว่า ①, ②

③ ดีสุด !! รู้ได้อย่างไร <u>ต้องพิสูจน์</u>

GreedyKnapsack (obj[1..n], W) {

$O(n \log n)$ sort obj[1..n] by $\frac{obj[i].v}{obj[i].w}$ nonincreasing

$\Theta(n)$ for (i = 1 to n) obj[i].x = 0;

$\Theta(1)$ i = ~~0~~ 1; sumW = 0

$\Theta(n)$ while (i <= n AND sumW < W) {

$\Theta(1)$ Obj[i].x = min(1, $\frac{(W-sumW)}{Obj[i].W}$)

sumW += min(obj[i].W, (W-sumW))

i++

}

return obj

}

Knapsack : Proving Optimality

- กำหนดของ n ชิ้น เรียงลำดับตาม v_i/w_i

  $\rightarrow \quad \frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \frac{v_3}{w_3} \geq \dots \geq \frac{v_n}{w_n}$

- ให้ $(x_1, x_2, \dots, x_n)$ เป็นผลเฉลยจาก greedy..
- ให้ $(y_1, y_2, \dots, y_n)$ เป็นผลเฉลยใดๆ ที่ถุงไม่ขาด

<u>จะพิสูจน์ว่า</u> $\left(\sum x_i v_i\right) - \left(\sum y_i v_i\right) \geq 0$

$$\hookrightarrow \sum (x_i - y_i) v_i \geq 0$$

<u>ข้อสังเกต</u>	$x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n$
greedy ได้	$1, 1, \dots 1, \quad x_k, 0, \dots, 0$
	โดยที่ $x_k < 1$

	1	2	...	k-1	k	k+1	...	n
x	1	1	...	1	x_k	0	...	0

$$\sum_{i=1}^{k-1}(x_i-y_i)v_i + (x_k-y_k)v_k + \sum_{i=k+1}^{n}(x_i-y_i)v_i$$

$$= \sum_{i=1}^{k-1}(x_i-y_i)w_i\frac{v_i}{w_i} + (x_k-y_k)w_k\frac{v_k}{w_k} + \sum_{i=k+1}^{n}(x_i-y_i)w_i\frac{v_i}{w_i}$$

$$\geq \sum_{i=1}^{k-1}(x_i-y_i)w_i\frac{v_k}{w_k} + (x_k-y_k)w_k\frac{v_k}{w_k} + \sum_{i=k+1}^{n}(x_i-y_i)w_i\frac{v_k}{w_k}$$

$$= \sum_{i=1}^{n}(x_i-y_i)w_i\frac{v_k}{w_k}$$

$$= \frac{v_k}{w_k}\sum(x_i-y_i)w_i \geq 0$$

$$= \frac{v_k}{w_k}\left(\sum x_iw_i - \sum y_iw_i\right)$$

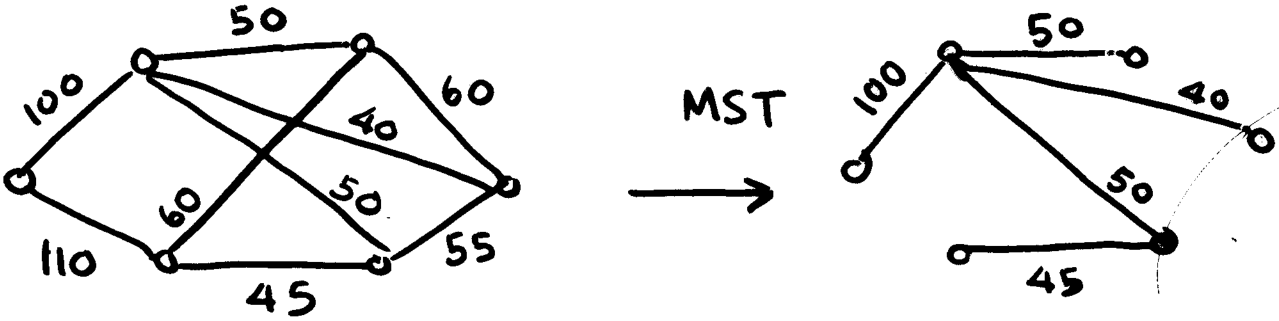
$$= \frac{v_k}{w_k}\left(W - \sum y_iw_i\right) \geq 0$$

$$\because \sum y_iw_i \leq W$$

$$\therefore \sum_{i=1}^{n}(x_i-y_i)v_i \geq 0$$

แสดงว่า $x_1, x_2, \ldots, x_n$ is optimal sol

Minimum Spanning Tree



Kruskal's algorithm

MST_Kruskal(G=(V,E)) {

$\Theta(1)$ F ← ∅

$O(e \log e)$ sort E by nondecreasing weight

$\Theta(e)$ for (each edge (u,v) ∈ E

in order by nondecreasing w) {

$\log^* e$ if noCycle(F ∪ {(u,v)})

F ← F ∪ {(u,v)}

}

return F

}

Implementation: ใช้ disjoint set

จะได้ $O(|E| \log |E|)$

Prim's Algorithm

MST_Prim($G = (V, E)$)

$E' \leftarrow E$

~~$e \leftarrow$ extract a shortest edge in E'~~

$T \leftarrow \cancel{\{e\}}$ เลือก ราก

while($|T| < |V| - 1$) {

$e \leftarrow$ extract a shortest edge in E'
which connected to T

if (noCycle($T \cup \{e\}$))

$T \leftarrow T \cup \{e\}$

}

}

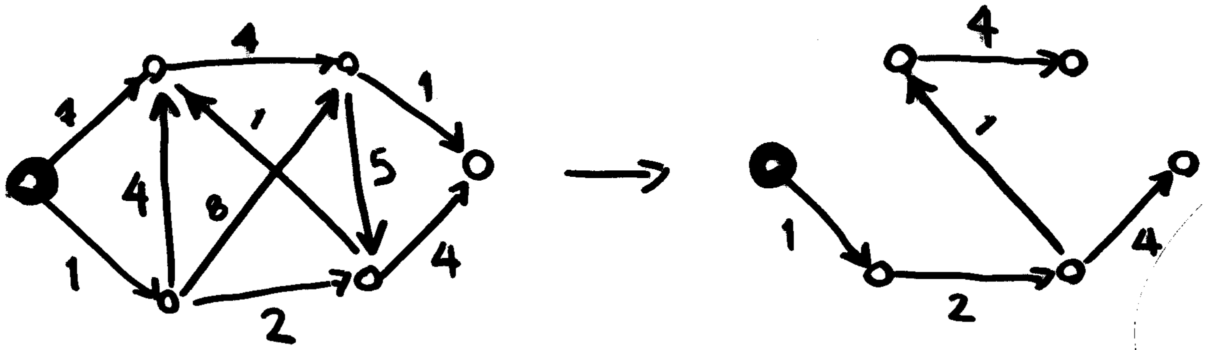
Implemention : ใช้ binary heap

ได้ $O(|E| \log |V| + |V| \log |V|)$

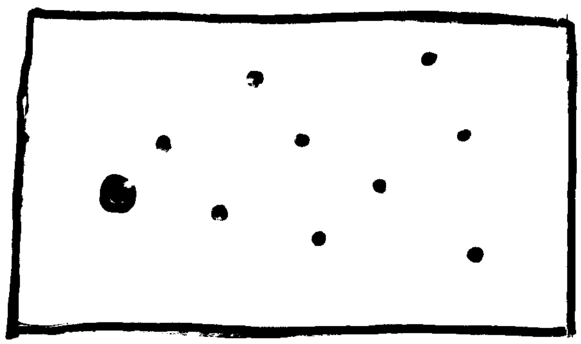
ใช้ Fibonacci heap

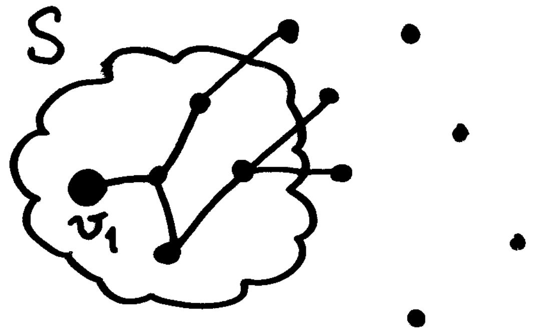
ได้ $O(|E| + |V| \log |V|)$

Single Source Shortest Path Problem



Dijkstra's Algorithm





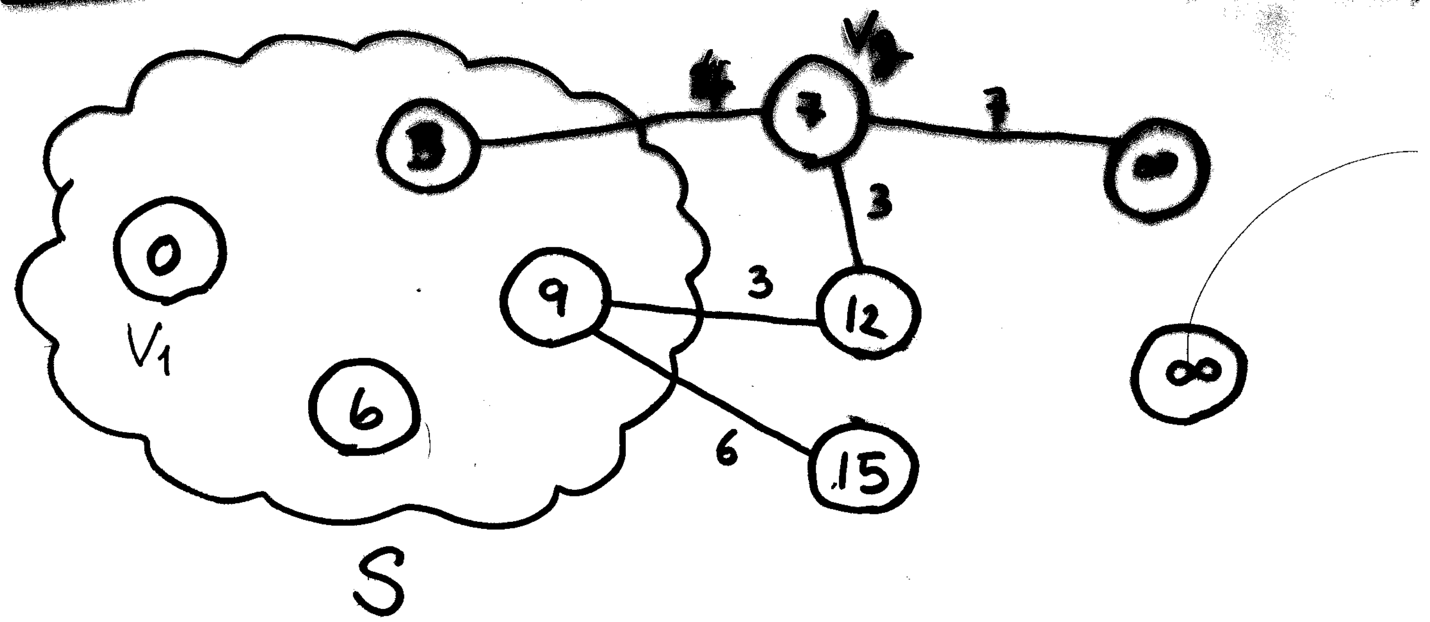
ให้ $\delta(v_i)$ คือ ระยะทางของวิถีสั้นสุดจาก $v_1 \rightarrow v_i$

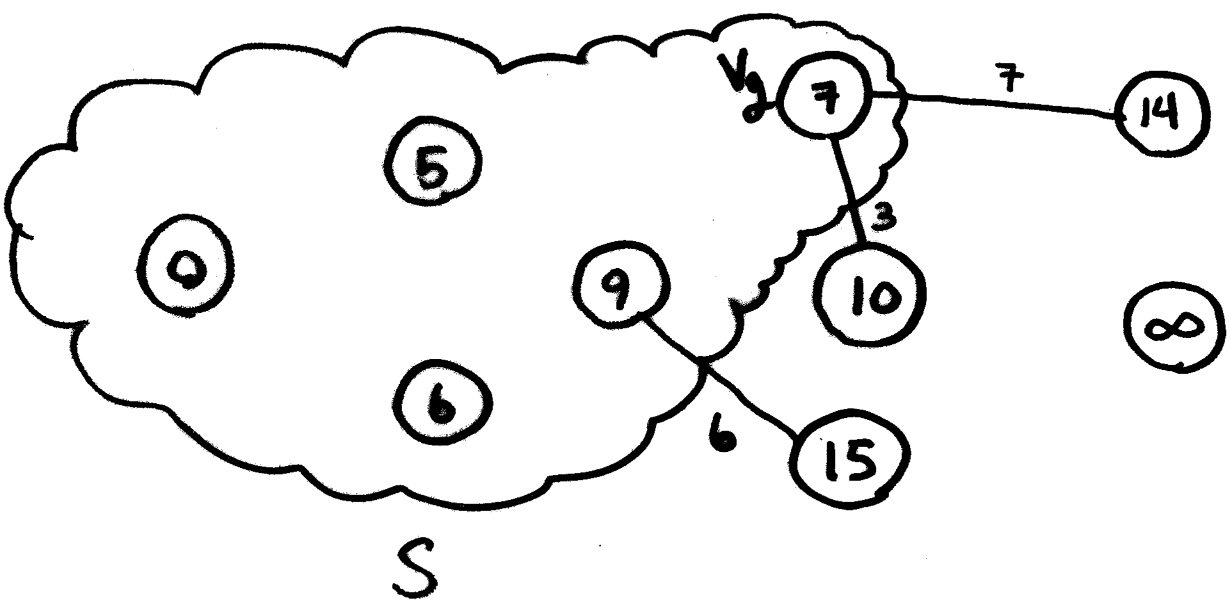
$d(v_i)$ คือ " ——— " ——— "

โดยผ่านจุดใน S เท่านั้น

$\Rightarrow \quad d(v_i) \geq \delta(v_i)$

claim: สำหรับ $v_i \in S$ $d(v_i) = \delta(v_i)$





for (each vertex $v_i \in adj(v_g)$)
relax (G, v_g, v_i)

relax (G, v_g, v_i)
if ($d(v_i) > d(v_g) + \omega(v_g, v_i)$)
$d(v_i) = d(v_g) + \omega(v_g, v_i)$

```
Dijkstra( G(V,E) v1 ) {
  for(each vertex vi ∈ V) d(vi) ← ∞
  d(v1) ← 0
  S ← ∅
  C ← a priority queue of V
        using d(vi) as priorities
  while ( C ≠ ∅ ) {
    vg ← extractMin(C)
    S ← S ∪ {vg}
    for(each vertex vi ∈ adj(vg))
        relax( G, vg, vi )
  }
}
```