#Q1

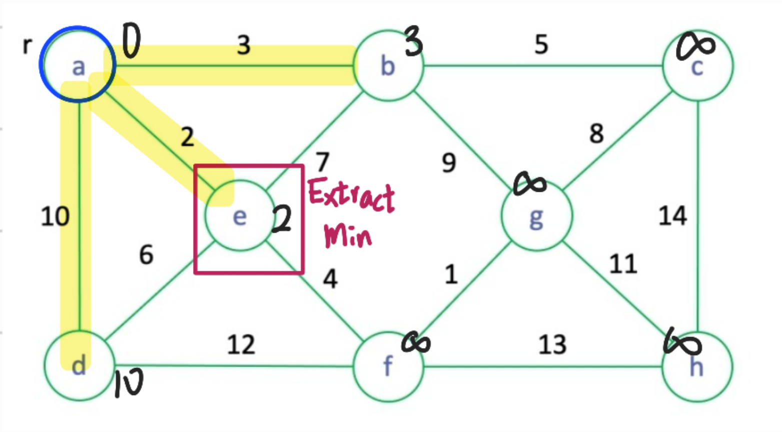
- Suppose that T is a minimum spanning tree of G including e, and let w_e be the weight of e. Then we can consider two cases below.

  1. There exists another edge whose weight is equal to w_e.
     - In this case, we can always make another spanning tree T', by excluding e and including the vertex v where $w_v = w_e$.
  2. Every other edge on the cycle has weight less than w_e.
     - In this case, let n be the number of vertices on the cycle of which e is an edge.
     - Then if we assume that every edge has non-negative weight, every minimum spanning tree connects the cycle with at most $n-1$ edges.
       - pf) Let T_0 be a minimum spanning tree that connects a cycle with n edges. Then since it is always possible to connect the vertices on the cycle with $n-1$ edges, and let T_1 be a spanning tree connected by those $n-1$ edges. Then the difference in the total weight of those trees, $W(T_0) - W(T_1) \geq 0$, which contradicts to the assumption that T_0 is a minimum spanning tree.
       - Thus we can say that every minimum spanning tree has at most $n-1$ edges if there is a cycle on a graph.
     - Therefore, there exists another tree T'' which connects the given cycle with $n-1$ edges, thus has less total weight than T. This contradicts to the assumption that T is a minimum spanning tree.
- As a result, we can say that there always exists another minimum spanning tree such that does not include e.

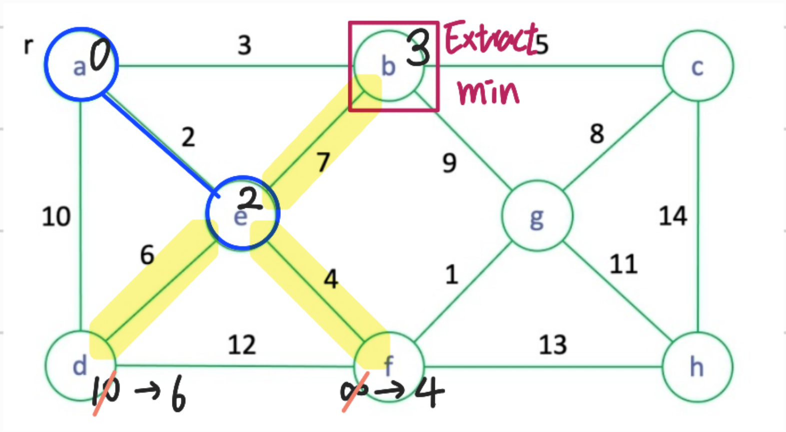
Q2

(a) Prim

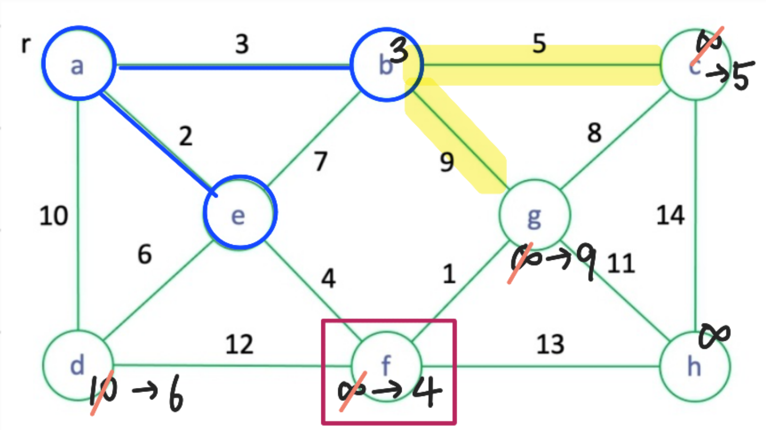
i) Step 1



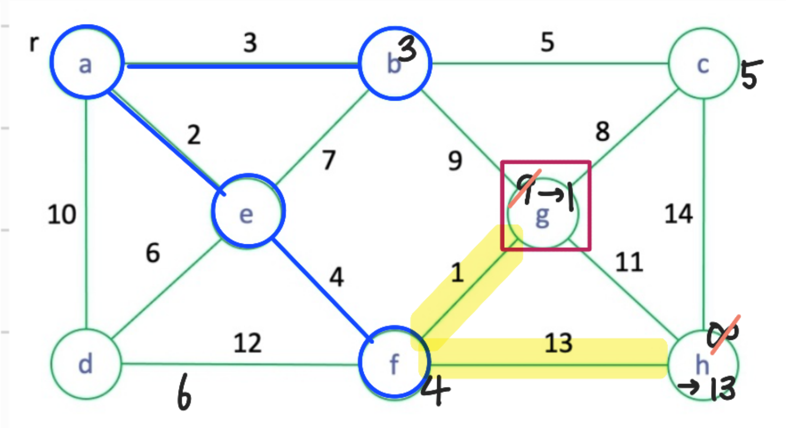
ii) Step 2



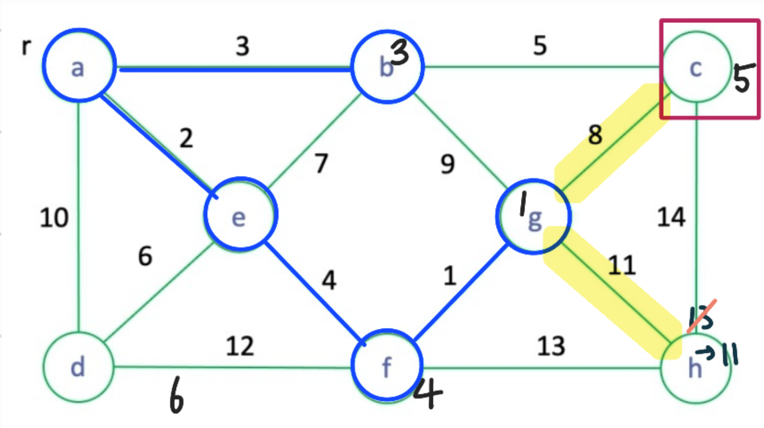
iii) Step 3



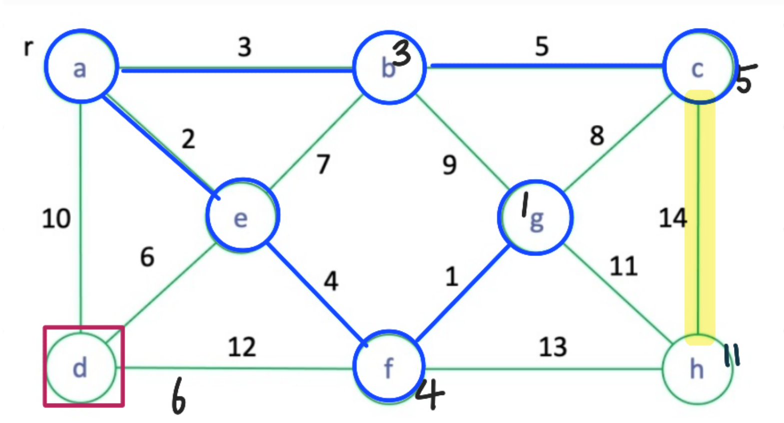
iv) Step 4



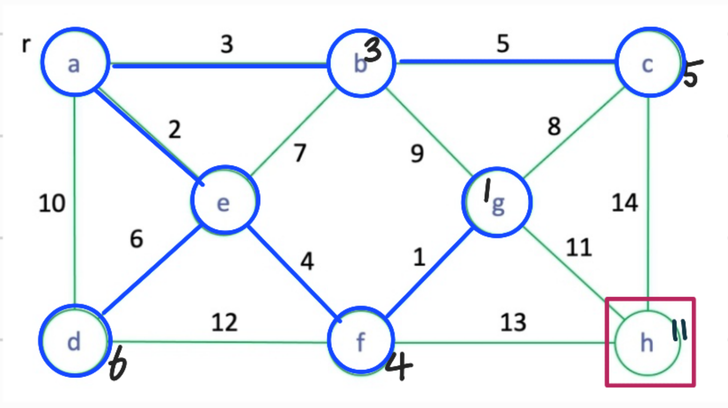
v) Step 5



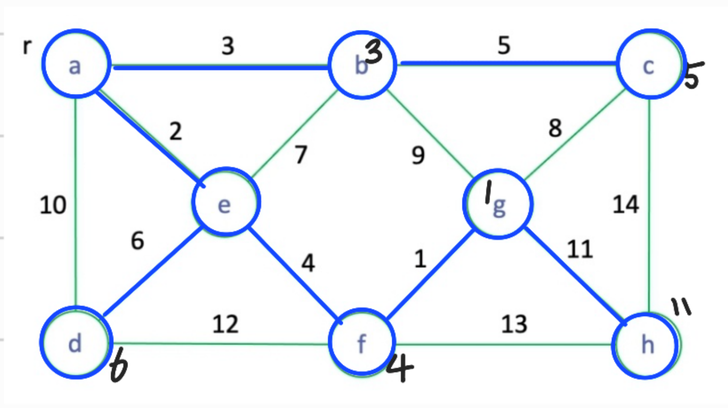
vi) Step 6



Vii) Step 7

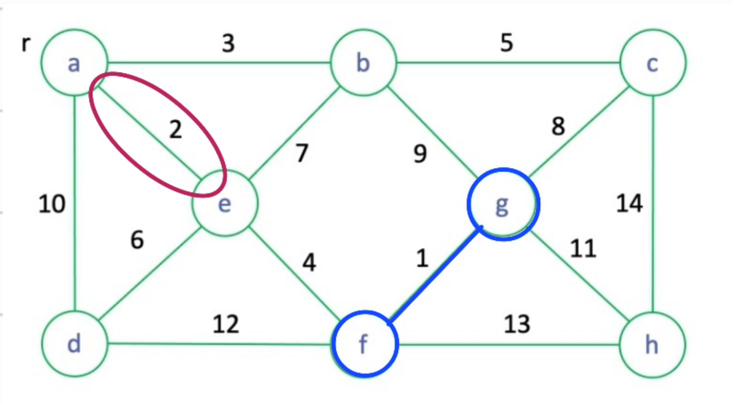


Vii) Step 8



(b) Kruskal

i) step 1



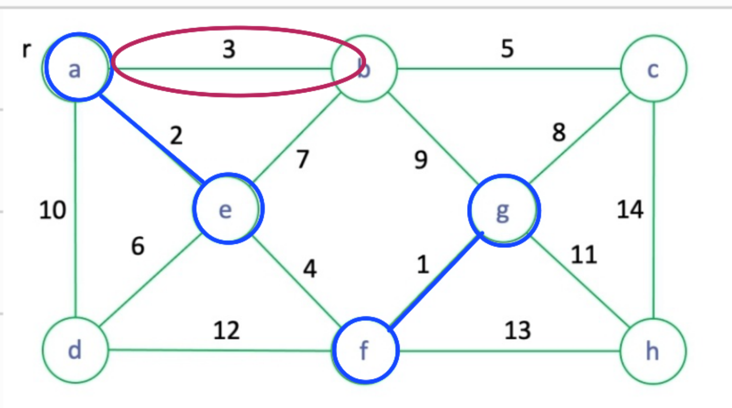
~ Edges sorted : {(f,g), (a,e), (a,b), (e,f), (b,c), (d,e), (b,e), (c,g), (b,g), (a,d), (g,h), (d,f), (f,h), (c,h)}

~ pop first edge (minimum-weighted edge) : (f,g)

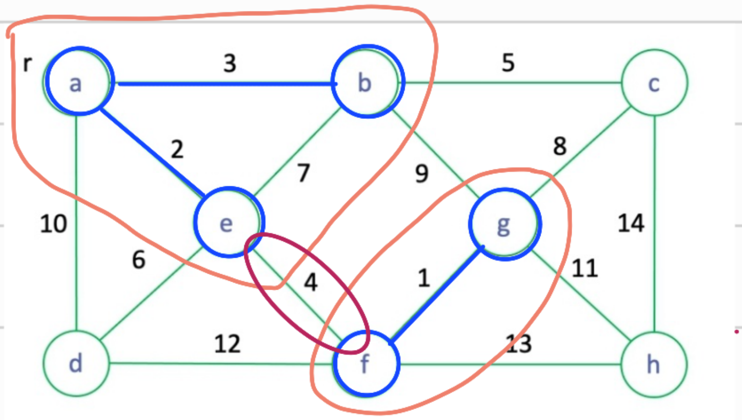
~ union g and f as one set

~ regard {g,f} as one set, then pop out the minimum weighted edge, and union again

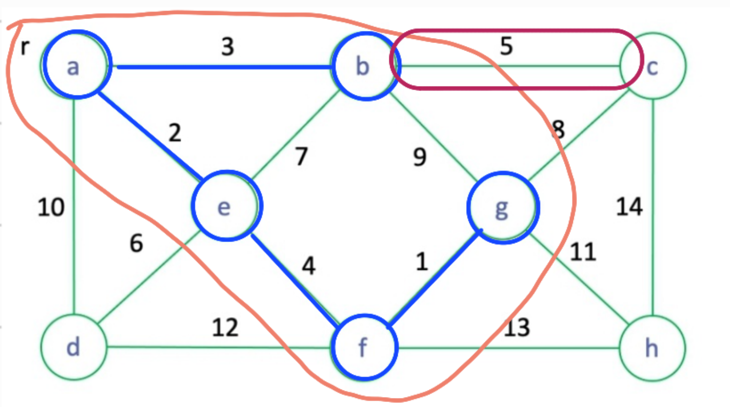
ii) Step 2



iii) Step 3



iv) Step 4

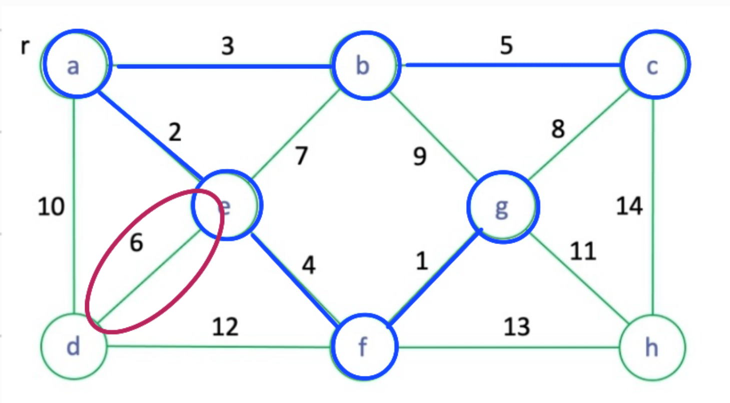


~ now, regard {a,b,e,f,g} as a set

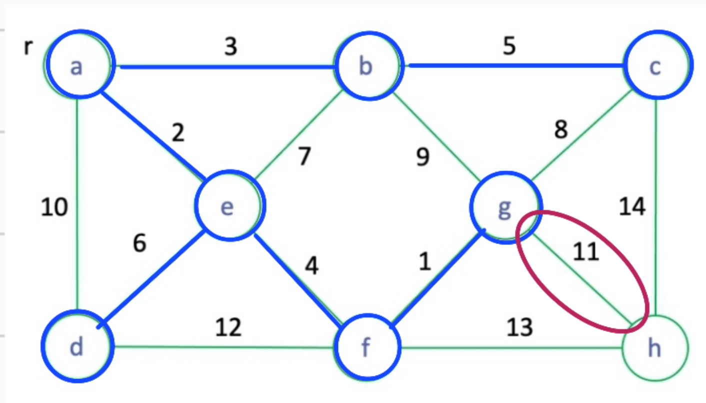
~ so the edge (b,e) and (b,g) are not considered any more

~ under this proposition, the minimum weighted edge is (b,c) now.

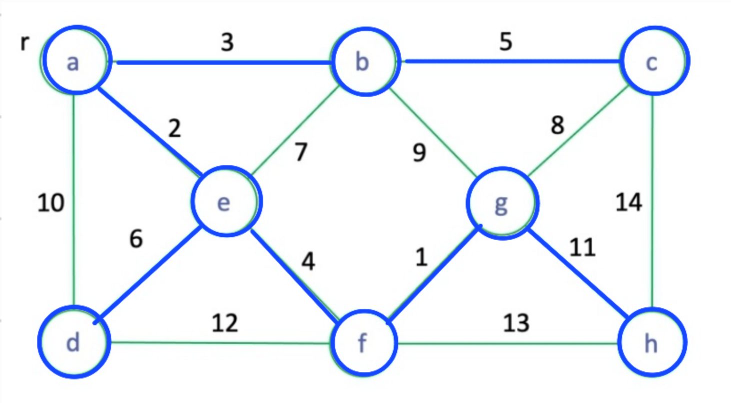
v) Step 5



vi) Step 6



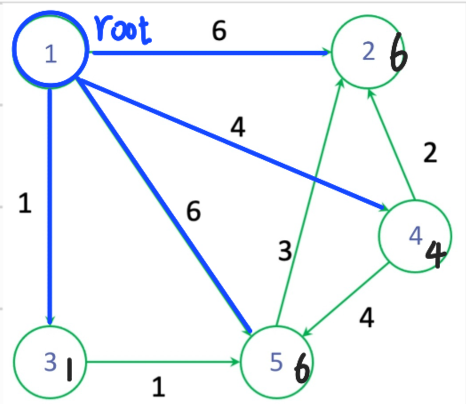
Vii) Step 7



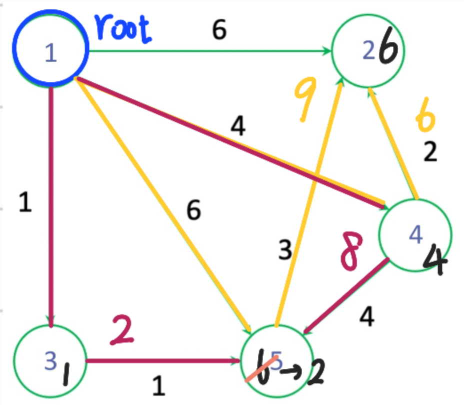
⇝ the final result is the same as the result from Prim's algorithm, while the result was achieved by taking one less step

Q3

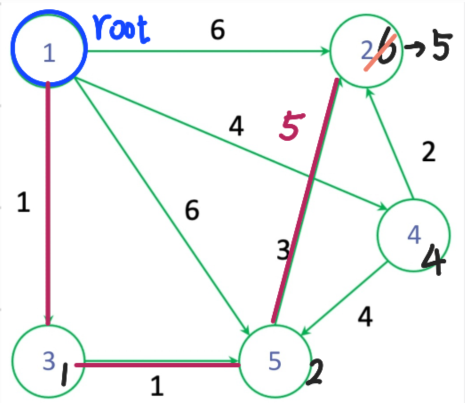
(a) i) Shortest path using 1 edge



ii) Shortest path using 2 edges



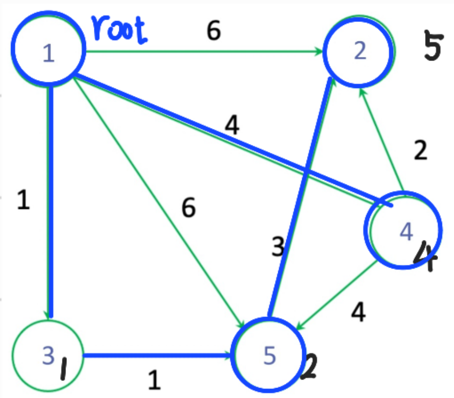
iii) Shortest path using 3 edges



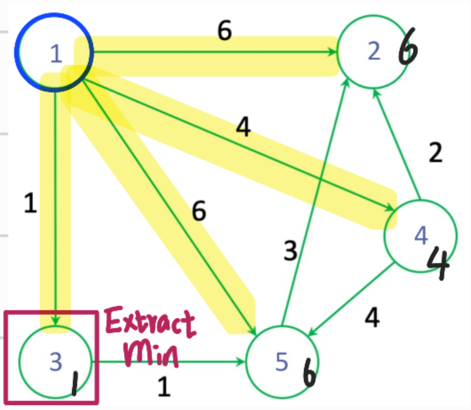
iv) shortest path using 4 edges

⇝ there is no path from the root using 4 edges

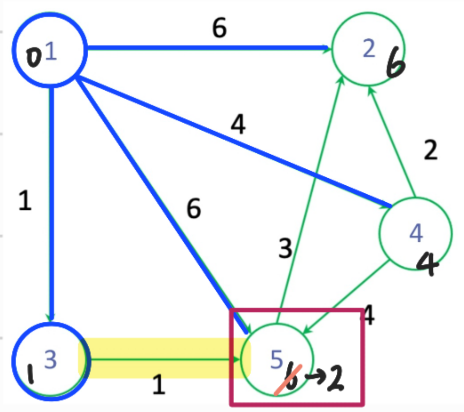
∴ The result is



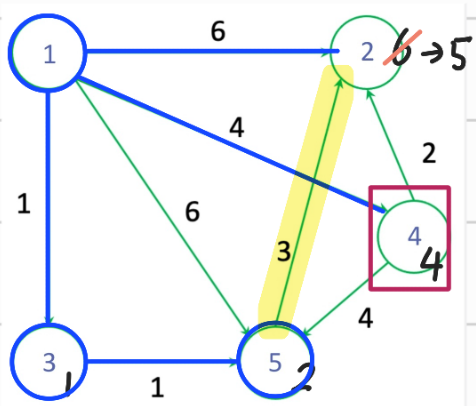
(b) i) Step 1



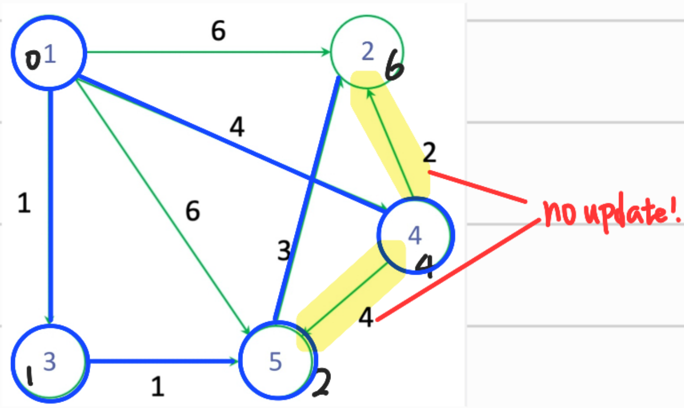
ii) Step 2



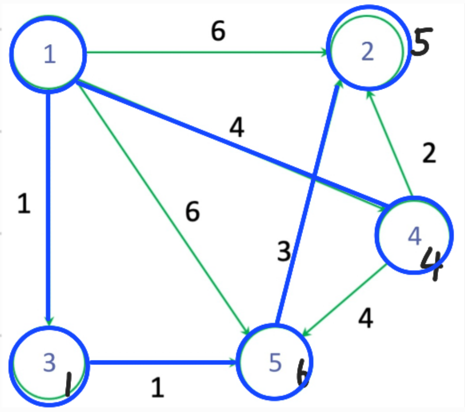
iii) Step 3



iv) Step 4



∴ final result



(c)
$$L^{(0)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array} \rightarrow L^{(1)} = W = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix} \end{array}$$

$$\rightarrow L^{(2)} = \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(1)}} \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{W}$$

↝ take an element-wise sum and set the $L^{(2)}_{11}$ as the minimum

$$= \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix} \end{array}$$

$$\rightarrow L^{(3)} = \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(2)}} \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{W} = \underbrace{\begin{pmatrix} 0 & 5 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(3)}}$$

$$\rightarrow L^{(4)} = \underbrace{\begin{pmatrix} 0 & 5 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(3)}} \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{W} = \underbrace{\begin{pmatrix} 0 & 5 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(4)}}$$

(d)

$$L^{(0)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array} \rightarrow L^{(1)} = W = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix} \end{array}$$

$$\rightarrow L^{(2)} = \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(1)}} \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{(1)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(2)}} \end{array}$$

$$\rightarrow L^{(4)} = \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(2)}} \underbrace{\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}}_{L^{(2)}} = \begin{pmatrix} 0 & 5 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$$

(e) $W^{(0)} = D^{(0)} = \begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$, $\Pi^{(0)} = \begin{pmatrix} NIL & 1 & 1 & 1 & 1 \\ NIL & NIL & NIL & NIL & NIL \\ NIL & NIL & NIL & NIL & 3 \\ NIL & 4 & NIL & NIL & 4 \\ NIL & 5 & NIL & NIL & NIL \end{pmatrix}$

⇝ Step $k=1$; if $d[i,1]+d[1,j] < d[i,j]$, update $d[i,j]$ ← no update

$\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$

1→1: 0   1→1: 0
1→2: 6   ~~2→1: ∞~~
1→3: 1   ~~3→1: ∞~~
1→4: 4   ~~4→1: ∞~~
1→5: 6   ~~5→1: ∞~~

} no update

$\Pi = \begin{pmatrix} NIL & 1 & 1 & 1 & 1 \\ NIL & NIL & NIL & NIL & NIL \\ NIL & NIL & NIL & NIL & 3 \\ NIL & 4 & NIL & NIL & 4 \\ NIL & 5 & NIL & NIL & NIL \end{pmatrix}$

⇝ step $k=2$: if $d[i,2]+d[2,j] < d[i,j]$ → update $d[i,j]$

$\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$

1→2 : 6   ~~2→1: ∞~~
2→2 : 0   2→2 : 0
~~3→2: ∞~~   ~~2→3: ∞~~
4→2: 2   ~~2→4: ∞~~
5→2: 3   ~~2→5: ∞~~

} no update

$\Pi = \begin{pmatrix} NIL & 1 & 1 & 1 & 1 \\ NIL & NIL & NIL & NIL & NIL \\ NIL & NIL & NIL & NIL & 3 \\ NIL & 4 & NIL & NIL & 4 \\ NIL & 5 & NIL & NIL & NIL \end{pmatrix}$

⇝ step $k=3$: $d[i,3]+d[3,j] < d[i,j]$ → update

$\begin{pmatrix} 0 & 6 & 1 & 4 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$

1→3: 1   ~~3→1: ∞~~
~~2→3: ∞~~   ~~3→2: ∞~~
3→3: 0   3→3: 0
~~4→3: ∞~~   ~~3→4: ∞~~
~~5→3: ∞~~   3→5 : 1

→ $\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$ $\Pi = \begin{pmatrix} NIL & 1 & 1 & 1 & 3 \\ NIL & NIL & NIL & NIL & NIL \\ NIL & NIL & NIL & NIL & 3 \\ NIL & 4 & NIL & NIL & 4 \\ NIL & 5 & NIL & NIL & NIL \end{pmatrix}$

⇝ 1→3→5 < 1→5 ; update $d[1,5]$

⇝ step $k=4$ → $d[i,4]+d[4,j] < d[i,j]$ → update

$\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$

1→4: 4   ~~4→1: ∞~~
2→4: ∞   4→2: 2
3→4: ∞   ~~4→3: ∞~~
4→4: 0   4→4: 0
5→4: 4   ~~4→5: ∞~~

→ $\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$ $\Pi = \begin{pmatrix} NIL & 1 & 1 & 1 & 3 \\ NIL & NIL & NIL & NIL & NIL \\ NIL & NIL & NIL & NIL & 3 \\ NIL & 4 & NIL & NIL & 4 \\ NIL & 5 & NIL & NIL & NIL \end{pmatrix}$

⇝ 1→4→2 : 6 vs 1→2 : 6
⇝ 5→4→2 : 6 vs 5→2 : 3
} no update

~> step $k=5$: $d[i,5] + d[5,j] < d[i,j] \longrightarrow$ update

$$\begin{pmatrix} 0 & 6 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix}$$

1→5: 2 ~~5→1: ∞~~
~~2→5: ∞~~ 5→2: 3
3→5: 1 ~~5→3: ∞~~
4→5: 4 ~~5→4: ∞~~
5→5: 0 5→5: 0

$$\begin{pmatrix} 0 & 5 & 1 & 4 & 2 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 4 & 0 & \infty & 1 \\ \infty & 2 & \infty & 0 & 4 \\ \infty & 3 & \infty & \infty & 0 \end{pmatrix} \quad \Pi = \begin{pmatrix} NIL & 5 & 1 & 1 & 3 \\ NIL & NIL & NIL & NIL & NIL \\ NIL & 5 & NIL & NIL & 3 \\ NIL & 4 & NIL & NIL & 4 \\ NIL & 5 & NIL & NIL & NIL \end{pmatrix}$$

$\boxed{\begin{array}{l} 1\to5\to2 : 5 \text{ vs } 1\to2 : 6 \\ 3\to5\to2 : 4 \text{ vs } 3\to2 : \infty \end{array}}$ → update

4→5→2: 7 vs 4→2: 4

Q4

```
DP_CUT_ROD(p, n):
  // p: an array containing the price of a rod cut at each point
  // n: the maximum length of a rod
  pi[0:n], s[1:n]
  // pi: an array containing the maximum profit by cutting the rod at each point
  // s: an array containing the cutting point to earn the maximum reward

  pi[0] = 0
  for j in (1:n): // j: the total length of a given rod
    q = -inf
    for i in (1:j): // i: each cutting point
      if (i == j): // if length equals to the cutting point -> no cutting!
        q = p[i] + r[j-i]
        s[i] = i
        pi[i] = q
      else:
        if q < p[i] + r[j-i] - c: // if p[i] + r[j-i] - c is the maximum profit
          q = p[i] + r[j-i] - c
          s[i] = i
          pi[i] = q

  return pi, s
```