Binary Search

$l \leftarrow 1$

$r \leftarrow n$

while $l <= r$.

$m \leftarrow \lfloor (l+r)/2 \rfloor$

if $k == A[m]$ return m.

else if $k < A[m]$ $r \leftarrow m-1$

else $l \leftarrow m+1$

return -1.

1 ← l

2

$\boxed{8}$ m

10

12

14 ← r

$$T(n) = T(n/2) + 1 \quad , T(1) = 1.$$

Multiplication of Large Integer

Multiply two n digit number it becomes $O(n^2)$. To reduce complexity use divide and conquer method.

$$C = a \times b.$$

$a = a_1 a_0$

$b = b_1 b_0$

$2442 = 24 | 42$ ($a_1$ = 24, $a_0$ = 42)

$1332 = 13 | 32$ ($b_1$ = 13, $b_0$ = 32)

$12 = 1 | 2$ ($a_1$ = 1, $a_0$ = 2)

$a = a_1 a_0$

$b = b_1 b_0$ [$\frac{n}{2}$ digit number]

$c_2 = a_1 \times b_1$

$c_0 = a_0 \times b_0$

$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$.

$c = c_2 * 10^n + c_1 \times 10^{n/2} + c_0$

eg: $a = 23$ $b = 14$.

$a_0 = 3$ $b_0 = 4$

$a_1 = 2$ $b_1 = 1$.

$c_0 = a_0 * b_0$

$= 3 * 4 = 12$

$c_1 = 5 \times 5 - 15$

$= 25 - 14$

$= 11$

$c_2 = a_1 \times b_1$

$= 2 \times 1 = 2$.

$c = 2 \times 10^2 + 11 \times 10 + 12$. $n = 2$

$= 200 + 110 + 12$.

$= 322$.

$\Theta(n^{\log_2 3})$. ⇒ Tym complexity

Strassen's Multiplication

$$A = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \quad B = \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix} \quad C = A * B$$

$$C = \begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix}$$

Matrix is $n \times n$ with $n$ as power of 2.

eg: $\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$

$$C = a * b = \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$

$m_1 = (a_{00} + a_{11}) * (b_{00} + b_{11})$.

$m_2 = (a_{10} + a_{11}) * b_{00}$

$m_3 = a_{00} * (b_{01} - b_{11})$

$m_4 = a_{11} * (b_{10} - b_{00})$

$m_5 = (a_{00} + a_{01}) * b_{11}$

$m_6 = (a_{10} - a_{00}) * (b_{00} + b_{01})$

$m_7 = (a_{01} - a_{11}) * (b_{10} + b_{11})$

eg:

$$\begin{array}{cc} a_{00} & a_{01} \\ \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix} \\ \begin{matrix} 1 & 3 \\ 1 & 0 \end{matrix} & \begin{matrix} 1 & 1 \\ 2 & 0 \end{matrix} \end{array}$$

1) $a = \begin{bmatrix} \overset{a_{00}}{1} & \overset{a_{01}}{2} \\ \overset{a_{10}}{3} & \overset{a_{11}}{4} \end{bmatrix}$ $\quad b = \begin{bmatrix} \overset{b_{00}}{2} & \overset{b_{01}}{1} \\ \overset{b_{10}}{1} & \overset{b_{11}}{1} \end{bmatrix}$

$m_1 = (a_{00} + a_{11}) \times (b_{00} + b_{11}) = 5 \times 3 = 15.$

$m_2 = (a_{10} + a_{11}) \times b_{00} = 7 \times 2 = 14.$

$m_3 = a_{00} \times (b_{01} - b_{11}) = 1 \times 0 = 0.$

$m_4 = a_{11} \times (b_{10} - b_{00}) = 4 \times (-1) = -4.$

$m_5 = (a_{00} + a_{01}) \times b_{11} = 3 \times 1 = 3$

$m_6 = (a_{10} - a_{00}) \times (b_{00} + b_{01}) = 2 \times 3 = 6.$

$m_7 = (a_{01} - a_{11}) \times (b_{10} + b_{11}) = -2 \times 2 = -4$

$$\begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5. \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$

Ans:

$$C = \begin{bmatrix} 4 & 3 \\ 10 & 7 \end{bmatrix}$$

Note:
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$ not possible its $n=3$ $\therefore$ $$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 4 & 5 & 6 \\ 0 & 7 & 8 & 9 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

n should be power of 2 $$\begin{bmatrix} 0 & 1 & 1 & 4 \\ 0 & 3 & 1 & 2 \\ 0 & 4 & 1 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$M(1) = 1$

$M(n) = \underbrace{7M(n/2)}$ → 7 multiplications

$O(n^{\log_2 7}) = O(n^{2.807})$

12 additions

6 subtraction

18

$7M(n/2) + 18n^2$

## Greedy approach.

At each step the choice is

* feasible (satisfy contrains)
* locally optimal
* irrevocable

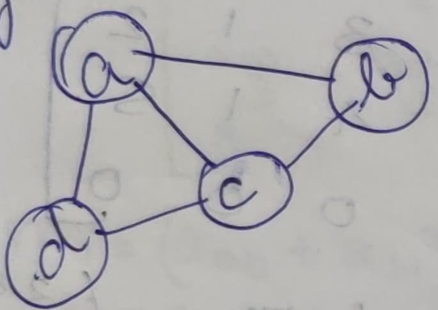
# Minimum Spanning Tree

From a graph form a spanning Tree

graph ↓ connected

spanning Tree ↓ connected acylic sub graph that contain all vertices of original graph.

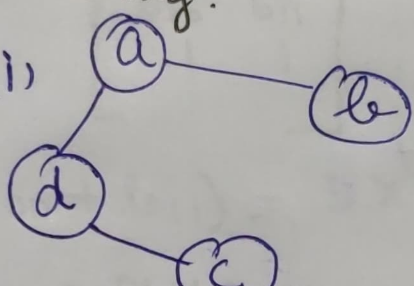
eg:



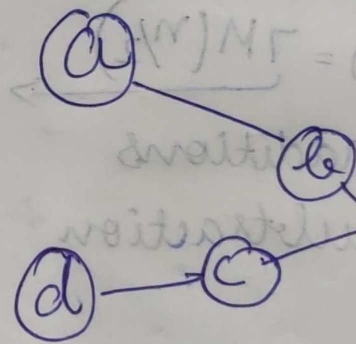
⟹

Spanning:

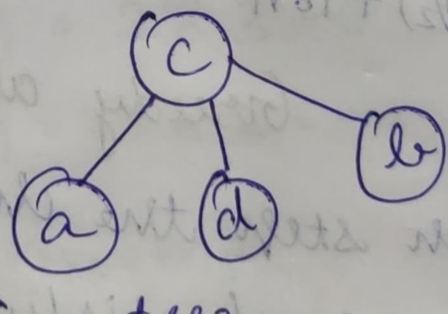
(i)



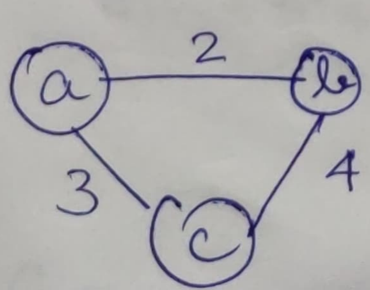
(ii)



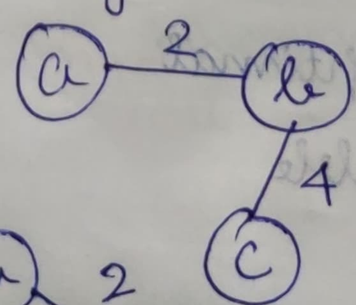
(iii)



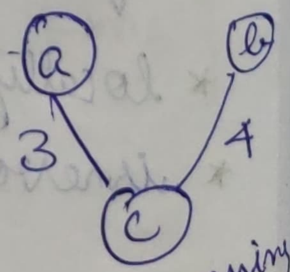
can't predict no. of spanning tree.

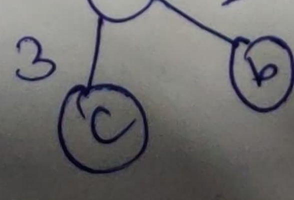


⟹



,





(3 are spanning tree)

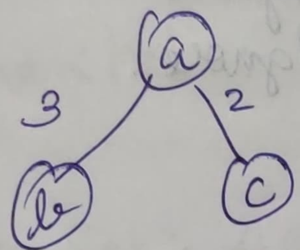
One can be only minimum spanning tree.

$2+4=6$

$3+4=7$. 5 is minimum.

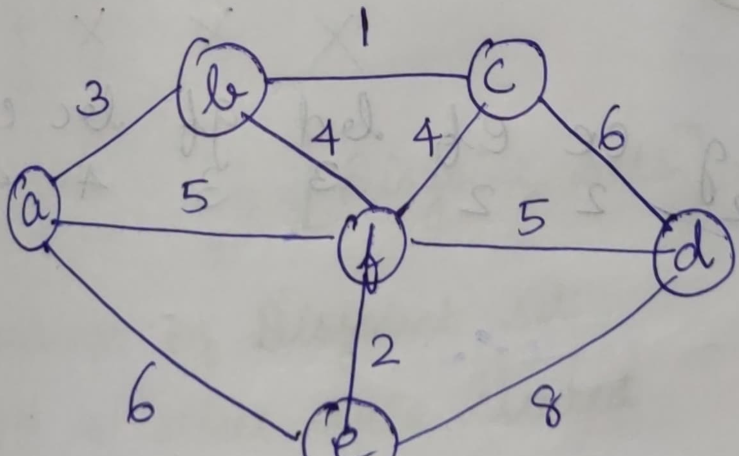
$2+3=5$.

$\therefore$ Minimum spanning tree is

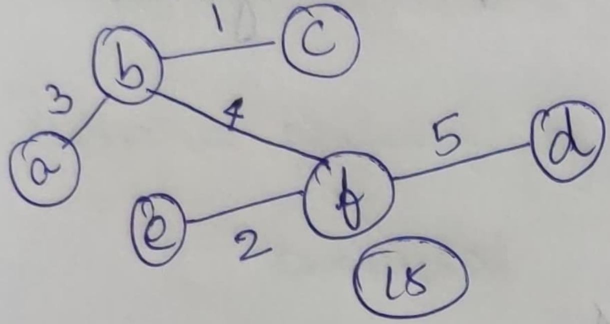



Kruskal's Algorithm.

Arrange edges in ascending order




| bc | ef | ab | bf | cf | af | fd | ae | cd | ed |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 4 | 5 | 5 | 6 | 6 | 8 |



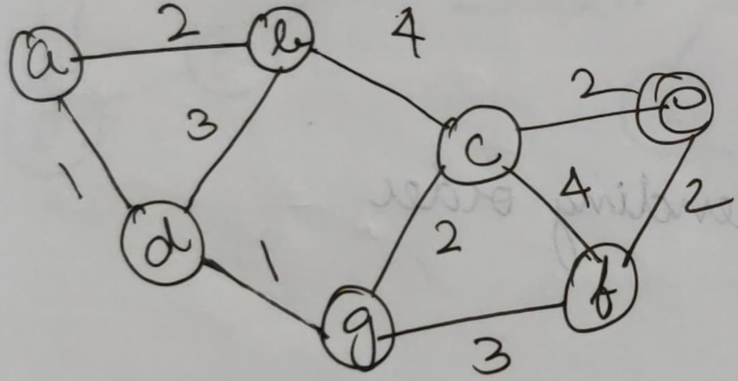

cf will make a cycle so just ignore.

af - Ignore.

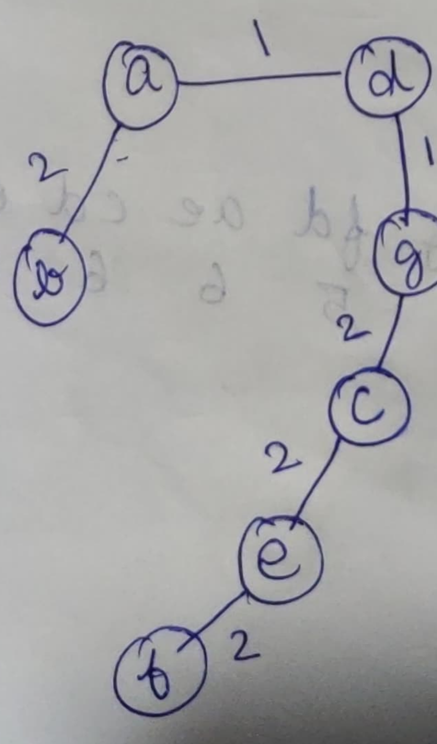
ae - Ignore

cd = Ignore

ed = Ignore.




| ad | dg | ab | cg | ce | ef | bd | gf | bc | cd |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 4 |




10 ⇒ sum of edges.

7... (c, c+2k

Krushal's Algorithm

Krushal($<V, E, W>$)

sort E so that $W(e_1) \leftarrow W(e_2) \leftarrow \cdots \leftarrow W(e_{|E|})$

$E_t \leftarrow \{\}$. $\longrightarrow$ Minimum spanning Tree

count $\leftarrow 0$.

$k \leftarrow 0$.

while count $< |V| - 1$

$k \leftarrow k+1$

if $E_t$ union $\{e_k\}$ is acyclic.

$E_t \leftarrow E_t$ union $\{e_k\}$.

count $\leftarrow$ count $+1$

return $E_t$.

ab de ef [Disjoint sets]
1 2 3

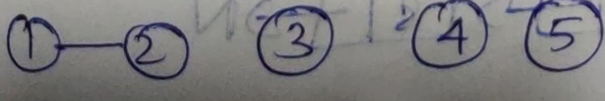
1. Creation of Disjoint set
2. Find whether cycle created
3. Union

$S = \{1, 2, 3, 4, 5\}$

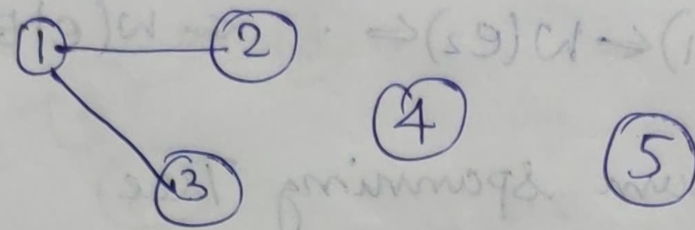
$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$

Make set.

Union $(1,2) \Rightarrow \{1,2\}, \{3\}, \{4\}, \{5\}$



Union[1,3] = {1,2,3}, {4}, {5}.



Union{2,3} × form cycle.

1st element of set is called representative

makeset(i).

{1}, {2}, {3}, {4}, {5}

can do union only if representatives are different

[1] → N (O(1))

[2] → N O(1)

U(1,2) [1] → [2]

U(1,3) [3] → [1] → [2] → N

8 elements (1-8)

[1] →
[2] →
[3] →
[4] →
[5] →
[6] →
[7] →
[8] →

Union (1,2)

[1 | -] → [2 | -] → N

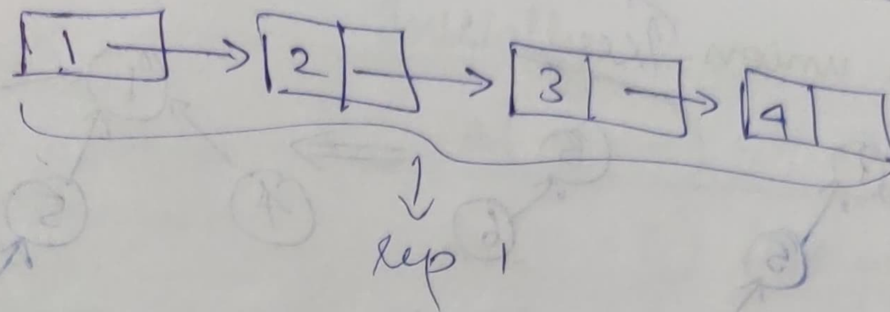
[3 | -] → [4 | -] → N

[5 | -] → [6 | -] → N

[7 | -] → [8 | -] → N

7.

U(1, 4)
↓ ↓
rep1 rep3
so perform union.




2's rep ⇒ updated 1

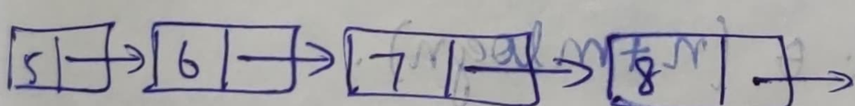
3's rep ⇒ updated 1 time

4's rep ⇒ updated 2 times (as 3, 1)

When there are 4 elements the max updation of an element is 2.

$n = 4$

$rep(up) = 2$ $\boxed{\log_2 4 = 2}$




$\boxed{\text{Linked List}}$

$n =$

$\log_2 n$

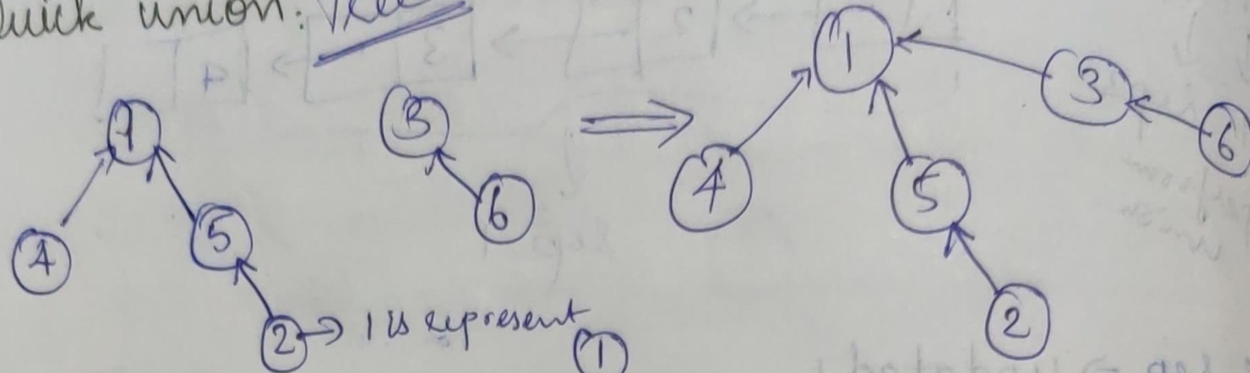
Union ⇒ $\Theta(n \log n)$.

$(n-1)$ union and $m$ find : $\Theta(m + n\log n)$

Two representation.

1. Linked List.
2. Tree

Quick union: Tree




| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | ⇒ Index of array |
| -1 | 5 | -1 | 1 | 1 | 3 | ⇒ array |

(-1) denotes itself as representative

Find: $\log n$ *
operation

* Makeset : $\theta(1)$.

$(n-1)$ union, $m$ find : $\theta(n + m \log n)$.

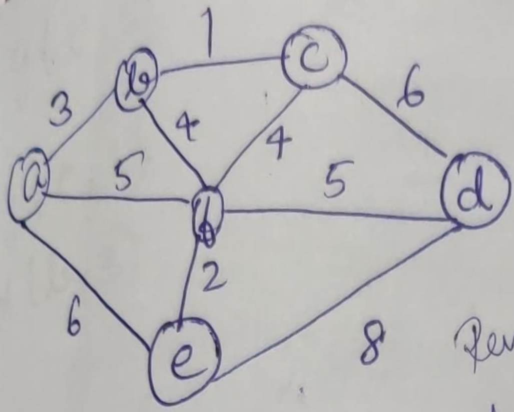
Union$(x,y)$ : $\theta(1)$.

Prim's Algorithm

a [vertex (in minimum, distance
spanning tree)
]

a ( – , – )

7.




Initially

(i) a(−, −) MST

b (a, 3) less weight

Remaining Vertex ⟹ c (−, ∞)

d (−, ∞)

e (a, 6)

f (a, 5)

(ii) MST

a(−,−) c(b, 1)

b(a, 3) d(−, ∞)

e(a, 6).

f(b, 4) because 4 < 5 (weight of b is less) so choose least weight.

MST Remaining.

(iii) a(−, −) d(c, 6).

b(a, 3) e(a, 6).

c(b, 1) f(c, 4) (choose either b or c)

(iv)

" d(f, 5)

f(c, 4). e(f; 2). ⟹ f(c, 4) d(f, 5)

e(f, 2).

(vi) a(−,−)

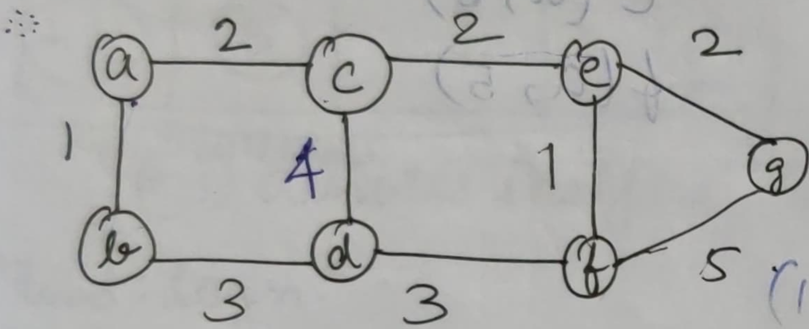
b(a,3)

c(b,1)

f(c,4)

e(f,2)

d(f,5)




a(−,−).   b(a,1), c(a,2), d(−,∞), e(−,∞)
f(−,∞), g(−,∞).

a(−,−)   c(a,2), d(b,3), e(−,∞), f(−,∞)
b(a,1)   g(−,∞).

a(−−)   d(b,3), e(c,2), f(−,∞), g(−,∞).
b(a,1)
c(a,2)

e(c,2).   d(b,3), f(e,1), g(e,2).

f(e,1).   d(b,3), g(e,2)

7.

g(e, 2) · d(b, 3)

g(e, 2) —

d(b, 3)

∴ a(−, −)

b(a, 1)

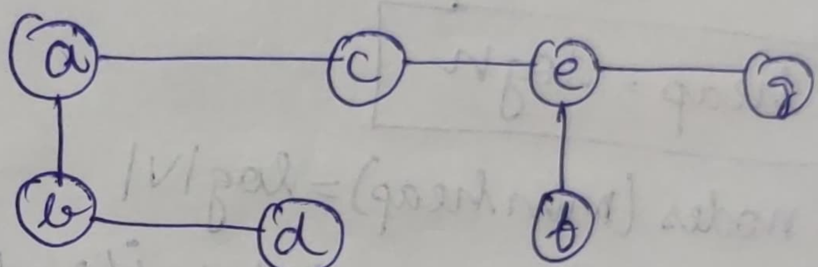
c(a, 2)

e(c, 2)

f(e, 1)

g(e, 2)

d(b, 3).



Remaing in vertex → priority queue as min heap.

Total vertex = |V|

In priority Queue total no. of Vertex = |V|-1

∴ |V|-1 deletion there in each iteration

$\boxed{\text{Min heap} = \log n}$

V nodes (min heap) = log|V|

(|V|-1 deletion + |E| Verification) log|V|.

$(|V-1| + |E|) \log |V|$

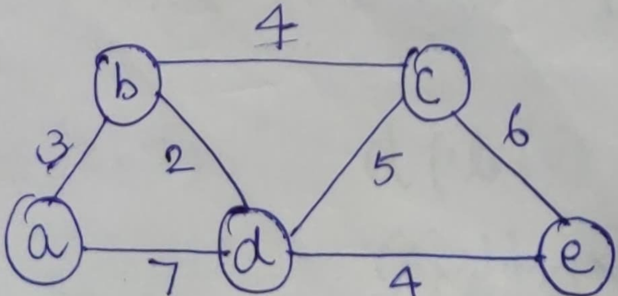
$|V-1| \leq |E|$

$|E| \log |V|$

Dijkstra's Algorithm

Single source shortest path.

* Not applicable for negative weight.
* Applicable for directed and undirected graph.




c (b, 7)

c from a through b.

3 + 4 = 7

a is the source vertex

tn.

Shortest Path.

| | |
|---|---|
| a(-, 0) | b(a, 3), c(-, ∞), d(a, 7) e(-, ∞) |
| b(a, 3) | c(b, 7), d(b, 5), e(-, ∞) |
| d(b, 5) | c(b, 7), e(d, 9). |
| c(b, 7) | e(d, 9) |

Adjacency matrix and priority queue as unordered array $\Theta|V^2|$

Adjacency lists and the priority queue as min-heap: $O(|E| \log |V|)$.

# Dynamic Programming
## (overlapping subproblem)

subproblem arise from recurrance relating a given problem's solution.

principle of optimality:

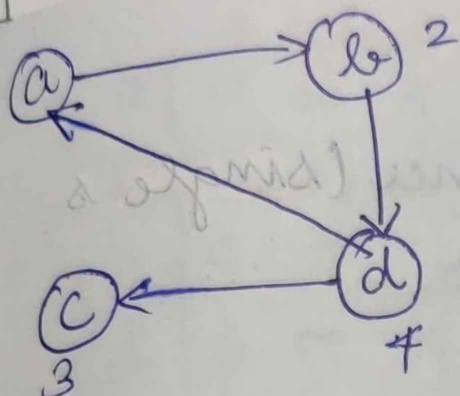
an optimal solution to any instance of an optimization problem is composed of optimal solutions to its sub-instances.

Warshall's Algorithm

Transitive closure of directed graph with n verten can be defined as the $n \times n$ boolean matrix.

(Whether there is a path or not (no matter how long (distance) it is))

* When there is a path from $i \rightarrow j$ then mark as 1.
* No path (b/w two points) 0.




$$A = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 0 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 0 & 1 & 0 \end{array} = R_0$$

$R_1$ (path matrix with intermediate ~~vertex~~ not >1) vertex

$$= \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 0 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 1 & 1 & 0 \end{array}$$

a and c has intermediate vertex as (b and d) but not a so leave it.

[either 0 or 'a' as intermediate vertex required]

b→b has path but (a, d) are intermediate.

$R^{2}$ = (not>2) path

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 1 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 1 & 1 & 1 \end{array}$$

$$R_3 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 1 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 1 & 1 & 1 \end{array}$$

$$R^4 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ b & 1 & 1 & 1 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 1 & 1 & 1 \end{array}$$

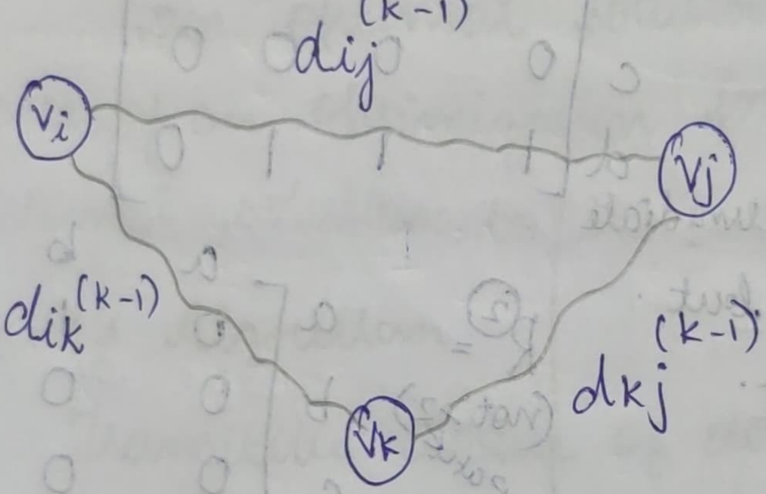
$$R^{n-1}_{ij} = R^{n}_{ij}$$

# Floyds Algorithm

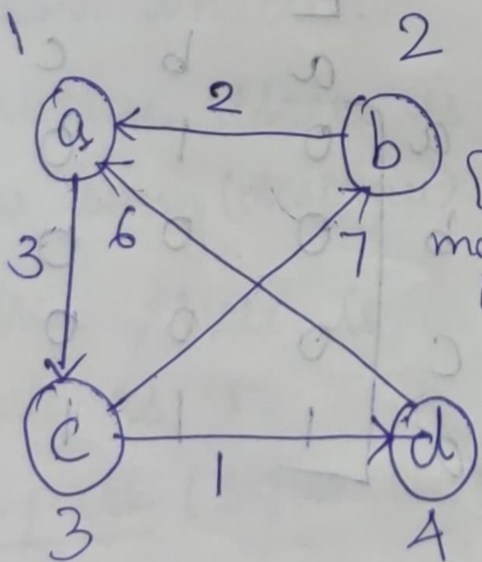
Find shortest distance

all pair (shortest path)

1, 2, 3, 4




$$\min\left(d_{ij}, \left(d_{ik} + d_{kj}\right)\right)$$




$D^0$ [distance matrix, 0 intermediate vertex]

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & \infty & 3 & \infty \\ b & 2 & 0 & \infty & \infty \\ c & \infty & 7 & 0 & 1 \\ d & 6 & \infty & \infty & 0 \end{array}$$

$D^1$ =

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & \infty & 3 & \infty \\ b & 2 & 0 & 5 & \infty \\ c & \infty & 7 & 0 & 1 \\ d & 6 & \infty & 9 & 0 \end{array}$$

[either 0 or 'a' alone intermediate]

$$D^2 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ 9 & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

$$D^3 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 9 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix} \end{array}$$

$$D^4 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 7 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix} \end{array}$$

for $k \leftarrow 1$ to $n$ do.
for $i \leftarrow 1$ to $n$ do
for $j \leftarrow 1$ to $n$ do.
$D[i,j] \leftarrow \min\{[D[i,j], D[i,k] + D[k,j]]\}$

Knapsack problem

W (capacity) weight of knapsack ⇒ +ive integer

Take it completely or don't take. (discrete)

Take it as a part (in continous)

$1 \rightarrow \overset{\text{(units)}}{w_1} \rightarrow v_1$ (value)

$2 \rightarrow w_2 \rightarrow v_2$

W ( Knapsack should be filled with valuable subset)

$$F(i,j) = \begin{cases} \max\{\overset{\text{may or may not include ith item}}{F(i-1,j)}, \overset{\text{include}}{v_i + F(i-1, j-w_i)}\} & j-w_i \geq 0 \\ F(i-1,j) & j-w_i < 0 \end{cases}$$

$v_i + F(i-1, j-w_i)$

$j = 10$

$w_i = 3.$

Remain ⇒ $j - w_i \Rightarrow 10 - 3 = 7$

$[1 - - - - i]$ elements.

(i) $i^{th}$ not included

(ii) may or maynot include $i$

$j$ = total Knapsack

| item | weight | Value |
|---|---|---|
| 1 | 2 | \$12 |
| 2 | 1 | \$10 |
| 3 | 3 | \$20 |
| 4 | 2 | \$15 |

$W = 5$

Approach

(i) Bottom up. (solve small-small sub-problem).

(ii) Top down. (Solve the goal).

as W=5, so coloumn = 5.

Row = 1 + no. of items.

$F(0,j) = 0$

$F(i,0) = 0$.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | F(1,1) 0 | (1,2) 12 | (1,3) 12 | 12 | 12. |
| 2 | 0 | 10 | 12 | 22 | 22 | 22. |
| 3 | 0 | 10 | 12 | 22 | 30 | 32. |
| 4. | 0 | 10 | 15 | 25 | 30 | 37 |

$\Rightarrow$ Ans = 37.

F(1, 1)

1 item given

$\rightarrow$ Knapsack = 1.

$F(1,1) \quad j - w_i = 1 - 2 < 0.$

$\quad\quad\quad\quad\quad \downarrow$

$\quad\quad\quad\quad$ weight, 1

$= F(i-1, j)$

$F(0, j)$

$F(1,2) = \quad j - w_i = 2 - 2 = 0 \geq 0.$

$= \max\{F(0,2), 12 + F(0,0)\}$

$= \max(0, 12)$

$= 12.$

$F(1,3) \quad j - w_i = 3 - 2 = 1 \geq 0.$

$\max(F(0,3), 12 + F(0,1)\}.$

$\max(0; 12)$

$= 12.$

$F(1,4) \quad j - w_i = 4 - 2 = 2 \geq 0.$

$\max(F(0,4), 12 + F(0,2)\}$

$\max(0, 12)$

$= 12$

$F(1,5) \quad j - w_i = 5 - 2 = 3 \geq 0.$

$\max\{F(0,5), 12 + F(0,3)\}$

$= 12.$

$F(2,1)$. $j - w_1 = 1 - 1 = 0. \geq 0$
$\downarrow$
$w$.

$\max(F(1,1), 10 + F(1,0))$

$\max(0, 10)$

$= 10$

$F(2,2)$. $j - w_1 = 2 - 1 = 1$.

$\max(F(1,2), 10 + F(1,1))$.

$\max(12, 10 + 0)$

$= 12$ //

$F(2,3)$ $j - w_i = 3 - 1 = 2$

$\max(F(1,3), 10 + F(1,2))$

$\max(12, 10 + 12)$

$= 22$.

$F(2,4)$ $j - w_1 = 4 - 1 = 3$.

$\max(F(1,4), 10 + F(1,3))$

$\max.(12, 10 + 12)$

~~max~~ $= 22$ //.

$F(2,5) = 22$.