25.03.2020 **Kruskal's Algorithm**

- The algorithm's state is modelled as a collection of disjoint sets, each of which contains the nodes of a particular component. ~~[illegible]~~
- Initially, each node is a component by itself.
  makeset(x): create a singleton set containing just x.
- We repeatedly test pairs of nodes to see if they belong to the same set.
  find(x): to which set does x belong?
- Whenever we add an ~~merging~~ edge, we are merging 2 components.
  union(x, y): merge the sets containing x & y.
- The algo uses |V| makeset, 2|E| find, and |V|-1 union operations.

PAGE NO.
DATE:

Algorithm

procedure kruskal $(G, w)$

Input: A connected undirected graph $G = (V, E)$ with edge weights $w_e$

Output: A min. spanning tree defined by the edges $X$.

```
for all u ∈ V:
    makeset(u)
X = { }
Sort the edges E in non-decreasing order of weight
for all edges {u,v} ∈ E, in increasing order of weight:
    if find(u) ≠ find(v):
        add edge {u,v} to X
        union (u,v)
```

One way to store a set is as a directed tree. Nodes of the tree are elts of the set, arranged in no particular order, and each has parent pointers that eventually lead up to the root of the tree.

Root elt: convenient representative / name of the set. Its parent ptr: self loop

Parent ptr : $\pi$

rank $r$: height of the subtree hanging from that node. (of a node)

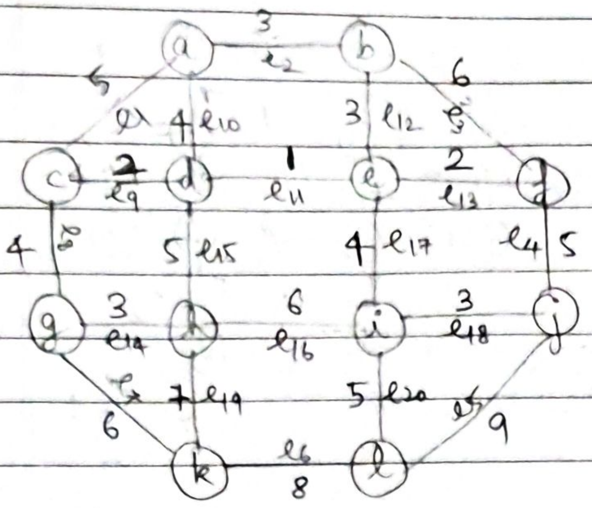
```
procedure makeset(x)
π(x) = x
rank(x) = 0

function find(x)
while x ≠ π(x): x = π(x)
return x

procedure union(x, y)
rx = find(x)
ry = find(y)
if rx = ry : return
if rank(rx) > rank(ry)
    π(ry) = rx
else:
    π(rx) = ry
    if rank(x) = rank(y): rank(y) = rank(y) + 1
```

Merging: make the root of the shorter tree point to the root of the taller tree.

Instead of explicitly computing hts of trees, we will use the rank nos. of their root nodes → union by rank

Use the algorithm to find min. cost spanning tree for the graphs given below:

(i)



|  | a | b | c | d | e | f | g | h | i | j | k | l |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| π: | a | b | c | d | e | f | g | h | i | j | k | l |
| rank: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E in non-decreasing order of wts:

$e_{11}, e_9, e_{13}, e_2, e_{12}, e_{14}, e_{18}, e_8, e_{10}, e_{17},$
1 2 2 3 3 3 3 4 4 4

$e_1, e_4, e_{15}, e_{20}, e_3, e_7, e_{16}, e_{19}, e_6, e_5$
5 5 5 5 6 6 6 7 8 9

① $e_{11}$ (d, e)

find(d) = d
find(e) = e
add (d, e) to x
union(d, e)
↳ $r_x = d$
$r_y = e$
$\pi(r_x) = r_y$ (d, e)

rank(d) = rank(e) ⇒ rank(e)++

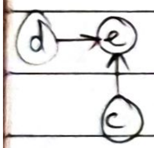
$x = \{\}$ $\{d, e\}$

| a | b | c | d | e | f | g | h | i | j | k | l |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | c | e | e | f | g | h | i | j | k | l |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

② $e_9$ (c, d)

$X = \{c, d, e\}$

union(c, d): $r_x = c$
$r_y = e$

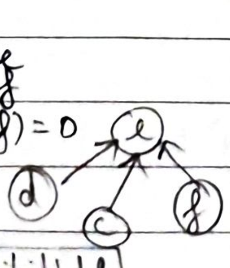


rank($r_x$) = 0 (c)
rank(e) = 1
⇒ ~~rank~~ $\pi(c) = e$
rank(e)++

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | e | e | e | f | g | h | i | j | k | l |
| 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

③ $e_{13}$ (e, f)

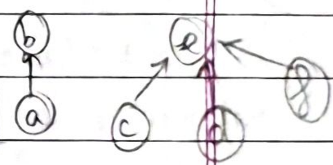
$X = \{c, d, e, f\}$
union(e, f): $r_x = e$
$r_y = f$
rank(e) = 2, rank(f) = 0
∴ $\pi(f) = e$.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | e | e | e | e | g | h | i | j | k | l |
| 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

④ $e_2$ (a, b)

$X = \{a, b, c, d, e, f\}$
union(a, b): $r_x = a$, $r_y = b$. rank(a) = 0, rank(b) = 0
∴ $\pi(a) = b$; rank(b)++



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | b | e | e | e | e | g | h | i | j | k | l |
| 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

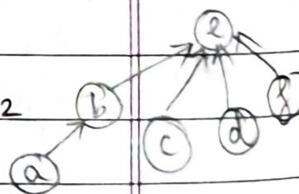
⑤ $e_{12}$ (b, e)

find(b) = b.
find(e) = e

$X = \{a, b, c, d, e, f\}$
union(b, e)

$r_x = b$, $r_y = e$. rank(b) = 1
rank(e) = 2

∴ $\pi(b) = e$; rank(e)++



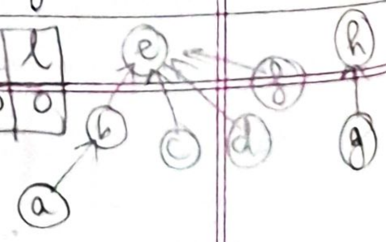
| a | b | c | d | e | f | g | h | i | j | k | l |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | e | e | e | e | e | g | h | i | j | k | l |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

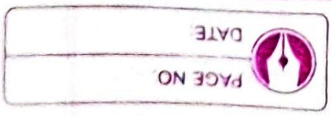
⑥ $e_{14}$ (g, h)

find(g) = g
find(h) = h

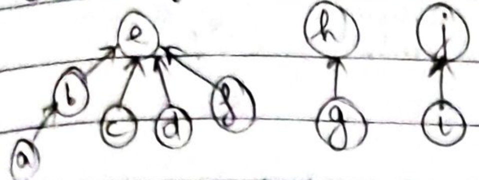
$X = \{a, b, c, d, e, f, g, h\}$
union(g, h) $r_x = g$, $r_y = h$.
$\pi(g) = h$ rank(h)++

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | e | e | e | e | e | h | h | i | j | k | l |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |





⑦ $e_{18}$ (i, j)

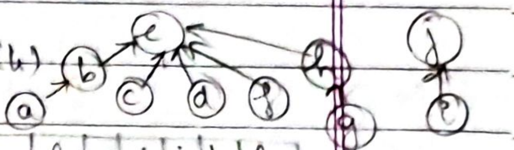


| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | e | e | e | e | e | h | h | j | j | k | l |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

⑧ $e_8$ (c, g) find(c) = e ← $r_x$
find(g) = h ← $r_y$

X = {a, b, c, d, e, f, g, h, i, j}

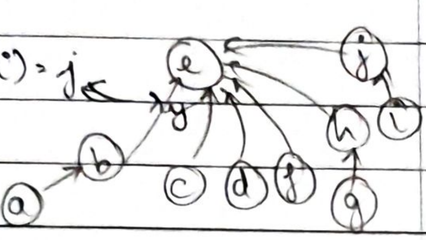
union(e, g)
rank(e) > rank(h)
∴ π(h) = e



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | e | e | e | e | e | h | e | j | j | k | l |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

⑨ $e_{10}$ (a, d)
find(a) = e, find(d) = e.

⑩ $e_{17}$ (e, i)
find(e) = e, find(i) = j
$r_x$
rank(e) > rank(j)

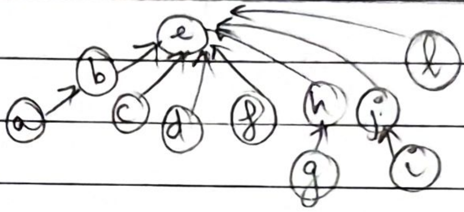


⑪ $e_1$ (a, c)
find(a) = e, find(c) = e.

⑫ $e_4$ (f, j)
find(f) = e, find(j) = e.

⑬ $e_{15}$ (d, h)
find(d) = e, find(h) = e.

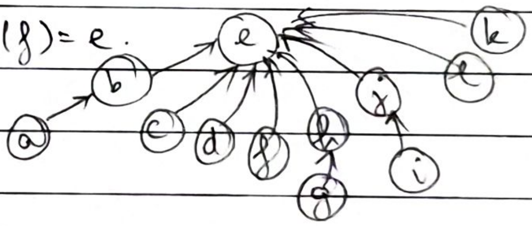
⑭ $e_{20}$ (i, l)
find(i) = e ← $r_x$, find(l) = l ← $r_y$.
X = {a, b, c, d, e, f, g, h, i, j, l}
rank(e) > rank(l)



| a | b | c | d | e | f | g | h | i | j | k | l |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | e | e | e | e | e | h | e | j | e | k | e |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

⑮ $e_3$ (b, f) → find(b) = find(f) = e.

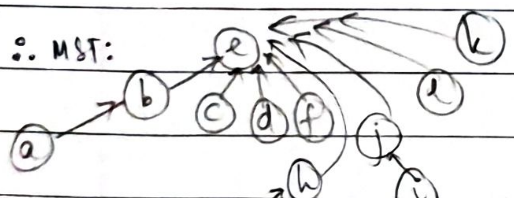
⑯ $e_7$ (g, k)
find(g) = e = $r_x$, find(k) = k.



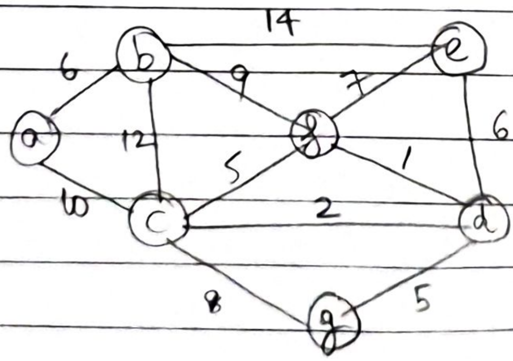
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| b | e | e | e | e | e | h | e | j | e | e | e |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

⑰ $e_{16}$ (h, i) ⑱ $e_{19}$ (h, k) ⑲ $e_6$ (k, l) ⑳ (j, l) $e_9$

∴ MST:



(ii)



| π: | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| rank: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E in non-decreasing order of wts.

(f, d), (c, d), (c, f), (d, g), (a, b), (d, e),
1 2 5 5 6 6

(e, f), (c, g), (b, f), (a, c), (b, c), (b, e)
7 8 9 10 12 14

(d,f)

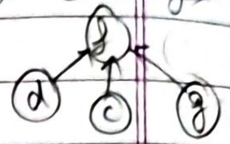
1 ~~(f,d)~~


| a | b | c | f | e | f | g |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 8 |

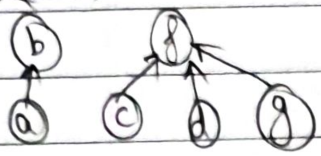
2 (c,d) find(c) = c
find(d) = f


3 (c,f)
creates cycle

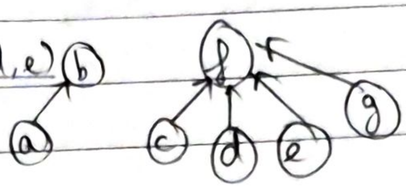
4 (d,g)




5 (a,b)



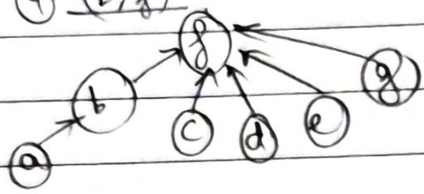

6 (d,e)




7 (e,f)
already present

8 (c,g)
will add cycle

9 (b,f)




∴ MST: