Kruskal's Algo

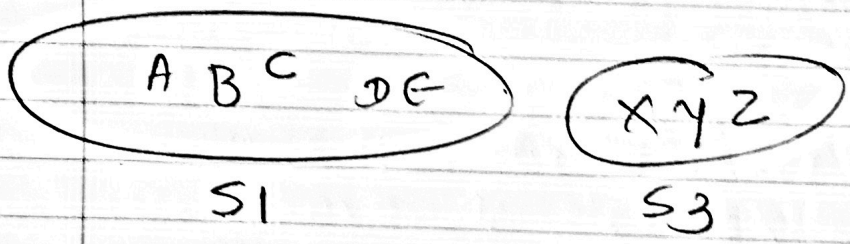
→ sorting edges in order of increasing cost
(rename edges 1,2,3 - - n so that $c_1 < c_2 < - - c_m$)

Disjoint set

two operations
- Union → joining two set into one set
- Find

find (B) = S1
find (D) = S1
find (X) = S3

ex:



Cycle defection

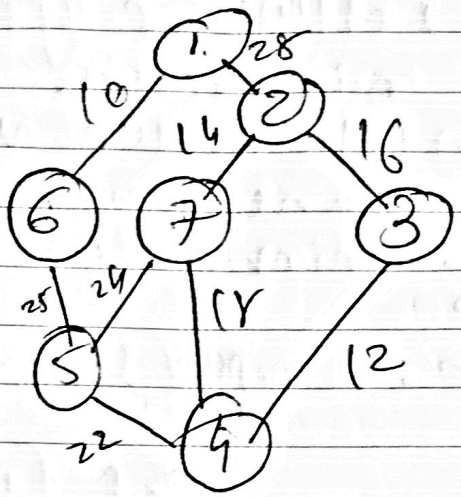
key point :- each edge
make subset
using the both vertices of the edge
if (both vertices are in same subset)
{
    a cycle is found
}

Psudo code:

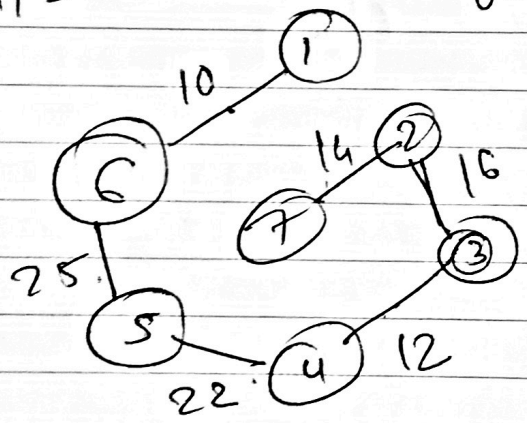
```
for each unvisited edge (u,v) in set E
{
    if [find(u) == find(v)]
    {
        // cycle is detected
    } else
        union(x, y);
}
```

in graph files → is cycle function

Kruskal's



Kruskal's smallest edge



→ always choose the minimum weighted edge but if it will make a cycle don't choose that edge.

→ so, (18) will make cycle without a cycle will not choose that - also don't included (24)

number of edges $|E|$

for spanning tree it will select same prim's algo

$|V| - 1$

$O(|V||E|)$

$O(n.E) = O(n^2)$

min-heap → always gives
minimum node
→ when we use remove
function
when we delete minnodes from
min-heap
time comp $O(\log n)$
↓
cost of finding
a minimum edge

so E = vertices − 1
so, $O(n.e)$
$O(n \log n)$
$O(E \log E)$