Kruskal's

↓

minimum spanning tree

↓

always select the minimum weighted edge from the graph

→ so, there are E number of edges into graph

→ so, for making spanning tree it will select , V - 1 = edges

V ↓ number of vertsises

$\therefore V - 1 = E$

→ runtime → $O(E.V)$ → time taken by Kruskul's algorithm

$O(E^2)$

→ min-Heap data structure comes in scenario

↳ Purpose is to storing all minimum weighted edges.

↳ and in min heap we know that the root node will be always minimum than other nodes.

If remove something from the min heap it will always remov min node

↳ for, our assign we are inserting the minimum weighted edges, removing the minimum weighted edge one by one.

→ we know, that algo for removing min node from the min heap is $\Theta(\log n)$

we can say that for finding the minimum weighted edge it will take $\Theta(\log n)$ time. $\Theta(\log |V-1|)$

$\Theta(\log E)$

→ so, from above Kruskal's algo

$\Theta(E \cdot V) \rightarrow V-1$ → too find the $V-1$ → edges it will take $\log n$ time.

$\Theta[E^2]$

$\Theta[E \cdot E]$

$\Theta(E \log E)$

→ So from this we can say that Kruskal's ~~algo~~ fastest possible runtime is $\Theta(E \log E)$.