Prim's $O(m+n \log n)$

Kruskal's Minimum Spanning Tree Alg

Also greedy alg.

$O(m \log m)$

Sorts all edges from smallest to highest weight.

$O(m)$

Nearly

Repeatedly add next to $T$ unless it creates a cycle ($O(m \cdot \text{inverse ackerman}(n))$)
Kruskal’s algorithm is done when \( n-1 \) edges in tree

See soon how to determine if there is a cycle in \( O(n+m) \) time
Data Structure called

Union-Find

1st edge

amortized cost for each union & find is inverse Ackerman function nearly constant
How does Prim's Alg avoid the sorting cost?

Each vertex (using priority queue) remembers its min weight connection does not sort all adjacent edges by weight.