Activity 3: Kruskal Simulation

for each edge in shortest order
add edge to MST if it doesn’t make a cycle
Activity 4: Runtime of Naïve Kruskal's
Fill in the left-hand run times below for Kruskal's Algorithm based on the naïve union-find implementation of merging clouds.

function kruskal(G):
  //Input: undirected, weighted graph G
  //Output: list of edges in MST
  for vertices v in G:
    makeCloud(v)
  MST = []
  Sort edges by weight
  for all edges (u,v):
    if u and v are not in same cloud:
      add (u,v) to MST
      merge clouds containing u and v
  return MST

Runtime of Naïve Kruskal's: ______________

Activity 5: Runtime of Path-Compression Kruskal's
Fill in the right-hand run times below of Kruskal's Algorithm based on the path-compression union-find implementation of merging clouds.

function kruskal(G):
  //Input: undirected, weighted graph G
  //Output: list of edges in MST
  for vertices v in G:
    makeCloud(v)
  MST = []
  Sort edges by weight
  for all edges (u,v):
    if u and v are not in same cloud:
      add (u,v) to MST
      merge clouds containing u and v
  return MST

Runtime of Path-Compression Kruskal's: ______________