We apply a modified version of Kruskal Algorithm shown as following:

Step 1. Assume there are m edges coming in total, and the number of vertices is n. For the first n-1 edges, we just store them into an array.

Step 2. For the next m-n+1 edges, starting from the n'th edge, put it into storage and each time apply Kruskal Algorithm, that is we sort the edges by value and choose n-1 edges with smallest values that are able to form a tree, i.e. MST. Each time we get a new MST, we delete the edge that is not in the tree from storage, since it cannot be in the MST anymore. It's obvious that we only store no more than n edges at any time, and after applying Kruskal Algorithm each time, we get an MST for the new graph. Generally take the storage of vertices, edges and the Union-Find Set into account, the total space used is in O(n).