# COMPLEXITIES:

**GREEDY ALORITHM:**

**HUFFMAN** with binary min heap: **O(n lg n).**

**HUFFMAN** with van Emde Boas tree : **O(n lg lg n).**

**GREEDY ACIVITY SELECTON**:

**DYNAMIC PROGRAMMING:**

//added by Shravya.

**ROD CUTTING: O(n^2)**

**LCS:** **O(m+n)**

**LEVENSHTEIN:** **O(m+n)**

**GRAPH:**

**MINIMUM SPANNING TREE (Chapter 23)**

**KRUSKAL:** **O(E lg V)**

when used path compression and union by rank.

Disjoint set operation for KRUSKAL takes O(E time. : initializing set A: O(1)

: sorting takes: O(E lg E)

: for loop that performs for FIND SET and UNION takes O(E)

: includes MAKE SET: O(V+E)

**PRIMS: O(E lg V)**

when we implement the priority Queue using the Min Binary Heap

When FIBONACCI HEAP is used the complexity becomes:

**O(E + V lg V )**

**SHORTEST SINGLE – SOURCE (Chapter 24)**

**BELLMANFORD: O(VE)**

**DAG :**

**DIJKSTRA’s :**