Optimal storage problem:

There are 'n' programs that are to be stored on a computer tape of length 'l'. Associated with each other i is the length li, 1<=i<=n. All the programs can only be written on the tape if the sum of all the lengths of the program is at most l.

Assumption: that whenever a program is to be retrieved from the tape, the tape is positioned at the front. Now if the programs are stored in the order as $I=i_1,i_2,i_3,i_4.....,in$, then the time t_j needed to retrieve program (i,j) is proportional to $\Sigma\ 1\leq k\leq j\ Li\leq 1$

. So the problem is that we have to store them in the tape in such an order that the M.R.T (mean retrieval time) is minimum.

Input: We are given 'n' problem that are to be stored on computer tape of length L and the length of program i is Li

Such that $1\leq i\leq n$ and $\Sigma\ 1\leq k\leq j\ Li\leq 1$

Output: A permutation from all n! For the n programs so that when they are stored on tape in the order the MRT is minimized.

Example:

Let n = 3, (l1, l2, l3) = (8, 12, 2). As n = 3, there are 3! =6 possible ordering.

All these orderings and their respective d value are given below:

Ordering	d (i)	Value
1, 2, 3	8 + (8+12) + (8+12+2)	50
1, 3, 2	8 + 8 + 2 + 8 + 2 + 12	40
2, 1, 3	12 + 12 + 8 +12 + 8 + 2	54
2, 3, 1	12 + 12 +2 +12 + 2 + 8	48
3, 1, 2	2 + 2 + 8 + 2 + 8+ 12	34
3, 2, 1	2 + 2 +12 + 2 + 12 + 8	38

(3, 1, 2) is optimal ordering

Optimal storage on multiple taps:

If we want to store files of lengths (in MB) {12,34,56,73,24,11,34,56,78,91,34,91,45} on three tapes. First sort the files in increasing order of length. For this we can use heapsort, meregesort or quicksort algorithms.

11	12	24	34	34	34	45	56	56	73	78	91	91

Now distribute the files: First element 11

Tape 1	11												
Tape 2													
Tape 3													

Second element 12

Tape 1	11												
Tape 2	12												
Tape 3													

Third element 24

Tape 1	11												
Tape 2	12												
Tape 3	24												

Finally we get,

Thirteenth element 91

Tape 1	11	34	45	73									
Tape 2	12	34	56	78									
Tape 3	24	34	56										

Minimum Cost Spanning Trees

Definition 1 Let $G = (V, E)$ be a connected undirected graph with weights on its edges. A *spanning tree* $(V; T)$ of G is a subgraph of G that is a tree. G is connected.

- If G is weighted and the sum of the weights of the edges in T is minimum, then $(V; T)$ is called a *minimum cost spanning tree* or simply a minimum *spanning tree*.
- If G is not connected, then the algorithm can be applied on each connected component of G.

Types of Minimum Cost Spanning Trees:

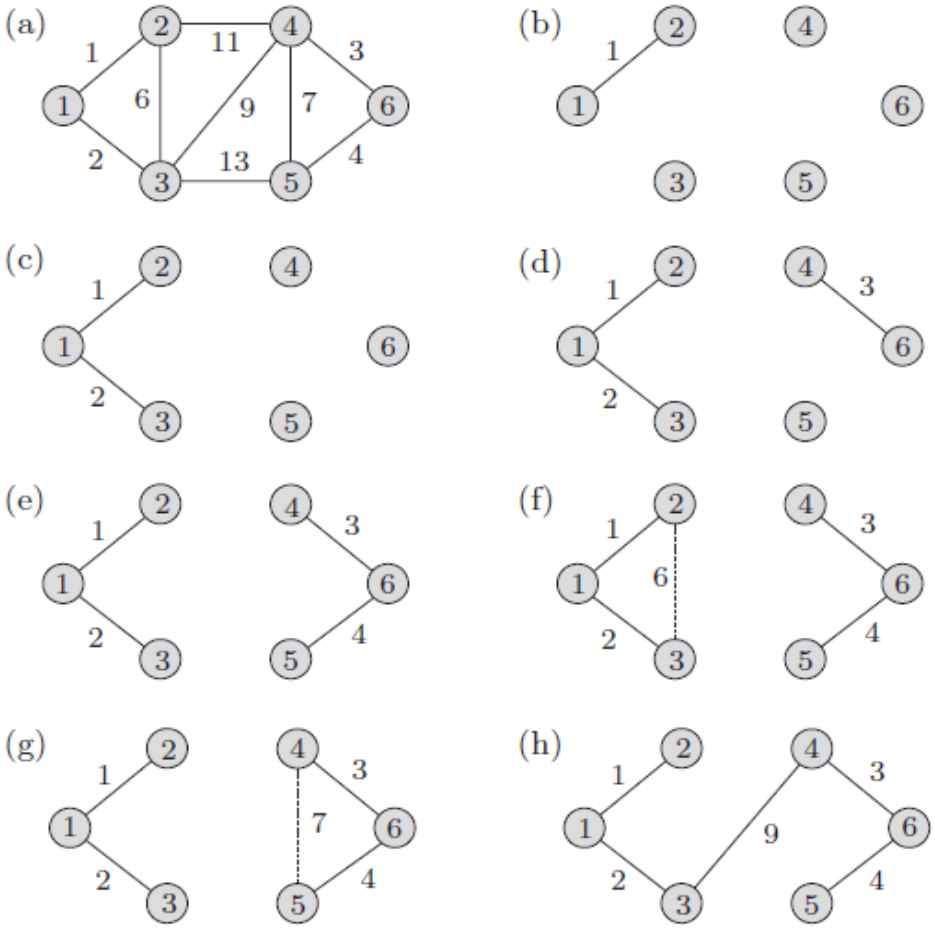
1. Kruskal's Algorithm
2. Prim's Algorithm

1.Kruskal's Algorithm

Kruskal's algorithm works by maintaining a forest consisting of several spanning trees that are. The algorithm starts by:

- Sort the edges in G by nondecreasing weight.

- For each edge in the sorted list, include that edge in the spanning tree T if it does not form a cycle with the edges currently included in T; otherwise discard it.
  Example: Show the result of applying Algorithm prim to find a minimum cost spanning tree for the undirected graph shown in Fig.



2. Prim's Algorithm

As in the previous section, we will assume throughout this section that *G* is connected.

- If *G* is not connected, then the algorithm can be applied on each connected component of *G*. This is another algorithm for finding a minimum cost spanning tree in a weighted undirected graph that has a totally undirected approach from that of Algorithm Kruskal's. Prim's algorithm. The algorithm grows the spanning tree starting:
  - Let G = (V, E) where for simplicity V is taken to be the set of integers {1, 2,… , n} The algorithm begins by creating two sets of vertices: X = {1} and Y = {2, 3,… n}. It then grows a spanning tree, one edge at a time.
  - At each step, it finds an edge (x, y) of minimum weight, where x ∈ X and y ∈ Y and moves y from Y to X.
  - This edge is added to the current minimum spanning tree edges in T. This step is repeated until Y becomes empty.

Example: Show the result of applying Algorithm prim to find a minimum cost spanning tree for the undirected graph shown in Fig.