CS 206M | Class Notes | 10th March

Dijkstra's algorithm - rectified :

Time required to go through the adjacency list is not E. It's ElogV because we're now using heaps and thus making changes in the heap is not constant time.
Time Complexity : O(Elog V + Vlog V)

If we use adjacency matrix instead of adjacency list we'll need more time as well as space.
Time Complexity : O(V^2 log V + Vlog V)

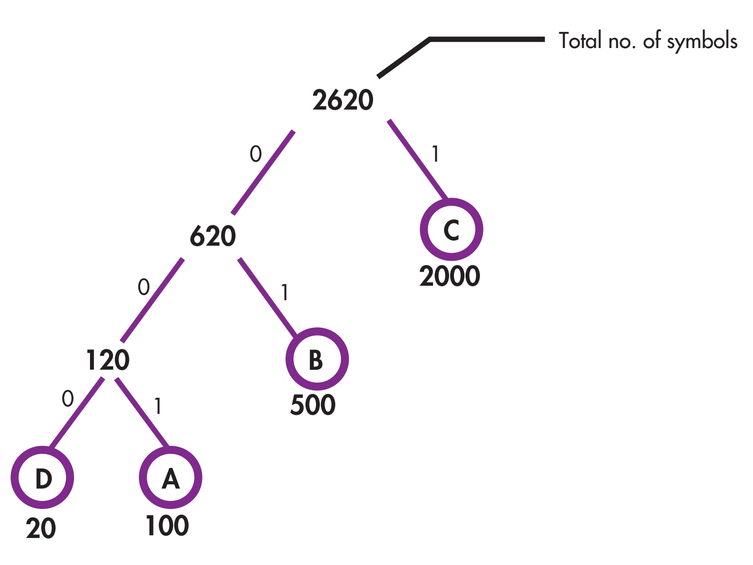
Huffman coding :

What :
1. Technique for data compression using binary codes.
2. It's based on the frequency of occurence of a data item. The principle is to use lower number of bits to encode the data that occurs more frequently.

How :
1. Sort the data according to the frequency.
2. Combine the two lowest frequencies to form a sub-tree.
3. Move the sub-tree to its correct place.
4. Use this logic recursively to get what is known as a Huffman Tree.

Example :
Suppose we have the following symbols in a particular text-

Symbol :	A	B	C	D
Frequency :	100	500	2000	20



Codes :

C	1
B	01
D	001
A	000

Thus, total no. of bits used : 2000 + 1000 + 300 + 60 = 3360

Without Huffman :
Each character takes 8 bits so total no. of bits : 2620*8

Therefore, we see we've achieved a considerable amount of compression :-)

Nota bene :
1. We choose two least frequency symbols at a time. Why don't we choose more than 2 ?
2. Huffman Codes are prefix free codes – Since all codewords are leaf nodes.
3. Decoding takes place at the server end.

Time Complexity :

Sorting - O(nlogn)
Inserting in the array - linear
T(n) : O(nlogn + n^2) = O(n^2)

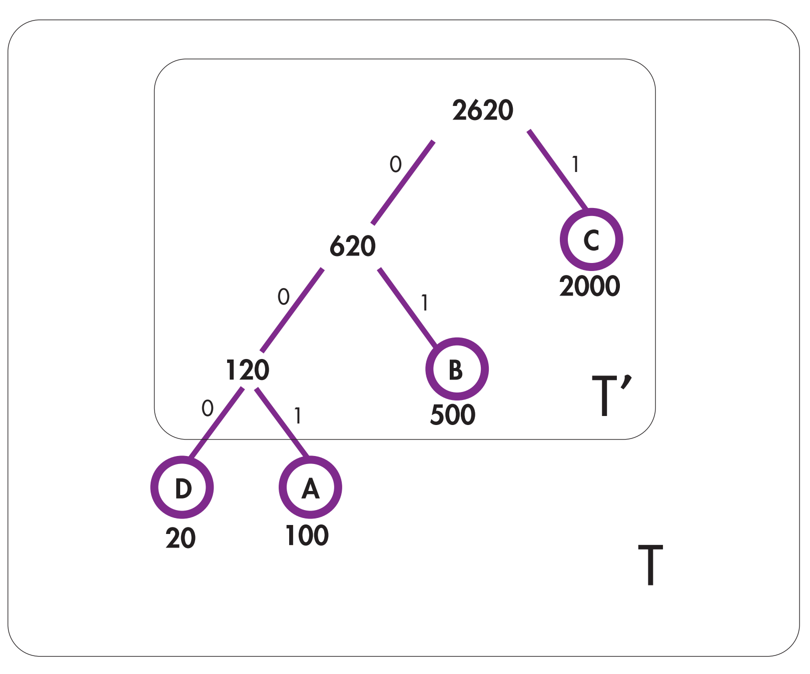
If we use Heap, we can reduce the complexity. Heap formation takes linear time and all the other operations take logn time.
T(n) : O(nlogn)

To Prove :

1. Given a set of symbols with frequency, there're many possible trees which give us optimal solutions. We get one out of them by using the greedy algorithm. We cannot get a non-optimal solution - the tree we get when using the greedy algorithm has cost same or less than any other tree.

Reason - In greedy solution we assign the longest code to the least frequency symbols. We place the least frequency symbols at the lowest point in the tree. If we replace them with any other (higher frequency) symbol, the cost is bound to increase.

2.



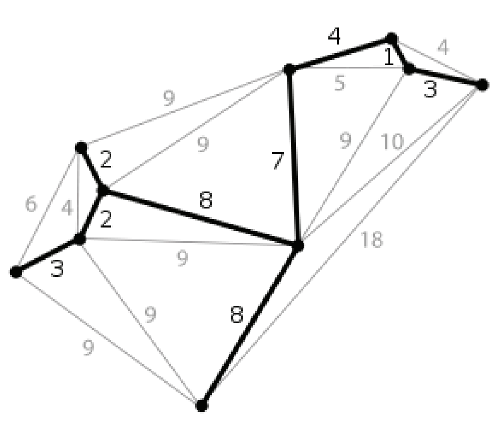
cost (T) = cost(T') + f(D) + f(A)
We've to prove that this is the minimum cost we've to pay for T.

Reason : As D and A are the symbols with lowest frequency, if we replace f(D) or f(A) with any other frequency, the cost will obviously be same or higher because all the other frequencies are either more or same as the frequency of A and D.

Minimum spanning Trees :

We're given an undirected graph G with positive edge weights (connected) and we've to find a minimum spanning tree.

Properties of MST :
1. It connects all the vertices.
2. Has no cycles.
3. The total weight should be minimum.

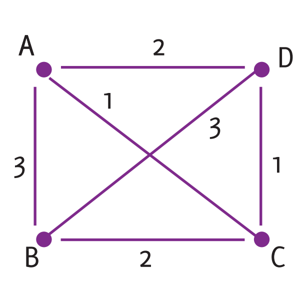


We'll discuss two greedy algorithms.

Algo 1 | Kruskal's algorithm :

1. We'll maintain an adjacency list.
2. Sort all the E edges. Consider the first edge - It'll be in T.
2. Add the next edge to T unless doing so would create a cycle.

	is Selected
A	yes
B	
C	yes
D	



AC	CD	AD	BC	AB	BD
1	1	2	2	3	3

1. Select AC. None of A or C is selected hence AC will be in the output.
2. Mark A and C as selected.
3. Repeat till no. of edges selected are V-1 (hence we've to maintain a count of the no. of edges selected too)

Time Complexity :

Sorting : ElogE
Other operations : constant
T(n) : O(ElogE)

N.B. Using heaps doesn't improve the asymptotic time complexity. It's still O(ElogE)

Algo 2 | Prim's algorithm :

Instead of discussing the algorithm we tried to design the algorithm from the very basic. Used the same concept of maintaining an adjacency list.
The challenge was how do we find the minimum edge (obviously which isn't already present in the MST) by looking at the adjacency list of all the selected edges. Thus, some other informations also need to be maintained.

Next class task : Try to design the algorithm - address all the problems we faced.

Thank You! ^_^

-Bhawna Agarwal
11020508