DAA ASSIGNMENT-6

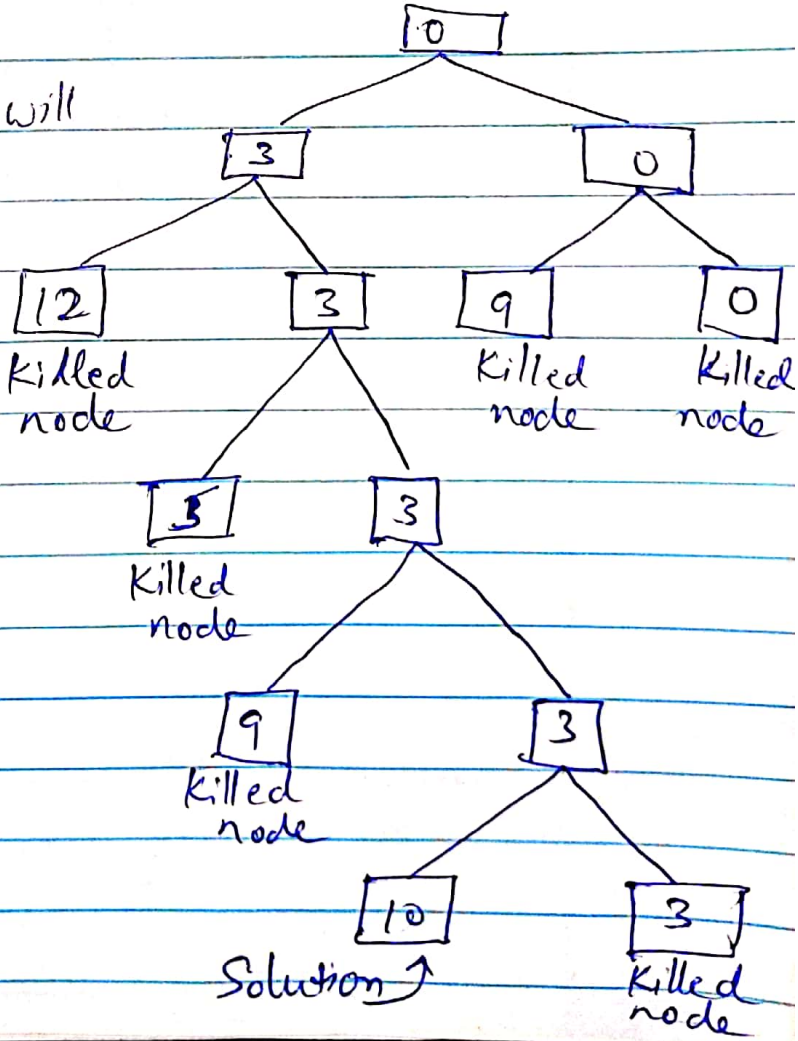
Submitted by -
BHUVAN CHADHA

1. Given Integers — 3, 9, 2, 6, 7.
Arranging in ascending order — 2, 3, 6, 7, 9.
Given Sum of Subsets, S = 10.

(a) If no pruning is done on the whole State space tree, then there will be 63 nodes in the tree.

(b) Pruned Tree

The prunned tree will Contain 13 nodes.



3. (a) The given problem belongs to the class P, that means, it can be solved in the polynomial time.

The array is sorted first, then the A[5000] element's value is compared with 10,0,000.

Sorting with Quick sort takes O(n log n) time. and the comparison of A[5000] with value 100,000 takes constant time.

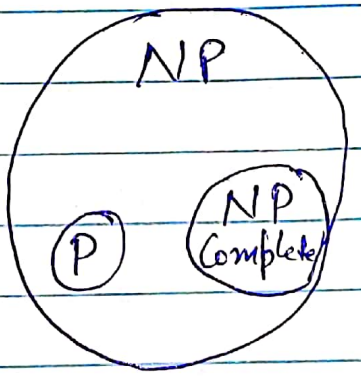
Hence, the problem can be solved in the polynomial time.

Algorithm

```
Sort A[]
if  A[5000] > 100,000
        return "Greater Element"
else if A[5000] < 100000
        return "Smaller Element"
else
        return "Equal"
```

(b) Since P class is a subset of NP class, i.e., polynomial time problems are a part of Non-Deterministic Polynomial time problems, the given problem ~~is~~ belongs to NP'.

(c) The given problem is both P and NP class problem. But, since NP-Complete is a subset of NP and need not necessarility be inclusive of P, it is not clear whether the problem belongs to NP-Complete or not. So, probably, it belongs to NP-Complete, but we can't be sure that it does.



4. (a) We know that problems of class P can be solved in polynomial time. Problems of class NP can be solved in Non-Deterministic Polynomial time. And NP-Hard problems are at least as hard as NP-Complete problems, which are the hardest problems to solve.

Now, $CC \leq_p$ NP Complete,
or, $X \leq_p Y$
This denotes that problem X can be reduced to another problem Y in polynomial

time. Or, it can be said that X is at least as hard as Y.
Therefore, problem CC is at least as hard as problems in NP-Complete

∴ CC is NP-Complete

Now, if any NP-Complete problem is in P, then P = NP.

If X is NP-Complete, then X is solvable in polynomial time if and only if P = NP.

Proof :
If P = NP, then X can be solved in polynomial time. Suppose X is solvable in polynomial time, and let Y be any problem in NP, then we can solve Y in polynomial time : reducing it to X.
Therefore, every problem in NP has a poly time algorithm and P = NP.

Thus, we can say, a problem Q is NP-Complete if -
(i) Q is NP
(ii) Every NP problem P reduces to Q.

Hence, from the above argument, we can say that Problem CC is in NP.

(2) No, ~~the~~ Professor Green should not get prize for showing P=NP.

5. (a) We are given that professor Watson showed a polynomial reduction of Satisfiability to the famous decision problem YY.
We know that that,
~~Sf~~ Satisfiability $\propto$ YY

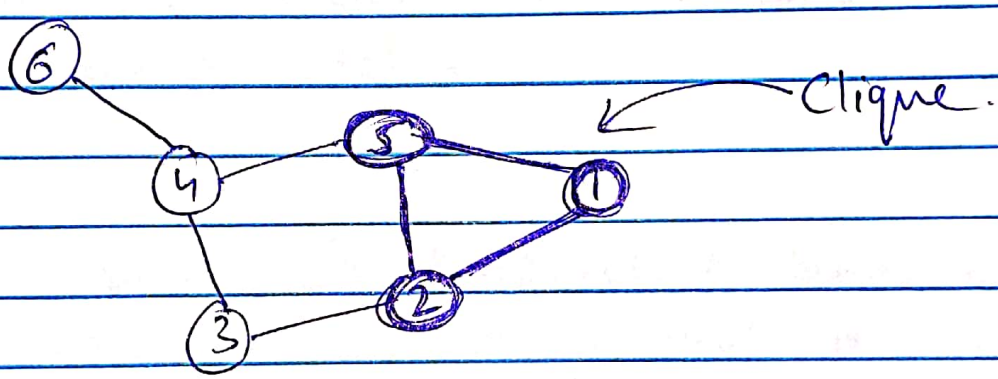
This is because Satisfiability is NP-Hard and YY is NP-Hard (reduction)

Now, if we reduce Satisfiability $\propto$ YY, it takes polynomial time. Hence, the professor knows that YY is an NP-Hard problem.

(b) Assuming that the professor finds a polynomial ~~time~~ bound algorithm for YY, it means that the problem YY is an NP-Complete problem as it can be solved

in polynomial time. Hence, P=NP.
Therefore, professor Watson should get the prize.

6. Given problem is the decision clique problem to determine whether a graph G contains a clique of at least a given size k.
This is an NP-Complete problem.



Professor Bartlett's algorithm takes $O(n^k)$ time for this problem, which is a polynomial time algorithm.
We know, $P \subseteq NP$
Hence, professor has showed P=NP.

7. The Sum of Subsets Problem is decision problem. It includes two subproblems -

(i) Include the last element, run recursively

for $n = n-1$, $Sum = Sum - Set[n-1]$

(ii) Exclude the last element, and run recursively for $n = n-1$.

If any of the given subproblems are true, then the program returns true.

Hence, the problem is in NP and it is an NP-Complete problem.

8. Considering 0/1 Knapsack problem, with two inputs - an array and an integer. The array consists of n items with each item having a weight and its index value. Maximum weight of an item is, say, W
Now, for instance, there are 5 items, and maximum weight is 3. In binary representation, it is 0011.
The time Complexity in this case is -

$$T(n) = O(n * W)$$
$$\Rightarrow O(5 \times 3) \Rightarrow \underline{O(15)}.$$

If Array size is doubled, the time Complexity,

$$T(n) = O(10 \times 3) \Rightarrow \underline{O(30)}.$$

But when size W is doubled, it means the binary value is doubled, i.e.,
$W = 00100000$, which is 8-bits long.

Now, $T(n) = O(n * W)$

$\Rightarrow O(5 \times 40) \Rightarrow O(200)$.

This is an exponential increase in time.
Hence, 0-1 knapsack problem is NP-Complete problem.

9.

(a) The Smarandache function, S(K) Calculation.

This problem is an NP class problem because the time complexity is exponential.

(b) Seat selection problem.

This problem is a decision problem of class NP because a decision is made based on the likes and dislikes of the child.

(c) Door finding decision problem.

This problem involves choosing a direction and ~~searching~~ exploring that direction for the door. If the door is found it means search successful, if door is not found, the algorithm Backtracks.
Hence, this problem is a P problem.