UNIT- 4

Learning Objective

NP-Completeness

- Polynomial-time verification
- NP-Completeness and Reducibility
- NP-Completeness Proof,
- NP-Complete problems.
- Branch and bound
- Backtracking and n-Queen's Problem
- Proof of cook's theorem.

NP Completeness

NP Completeness

NP Completeness Cont...

- There are certain problem which can be solved in linear time and there are certain problem which can be solved in polynomial time.
- But there is a special class of problems which include such problems for which algorithms with polynomial Running times have not be designed these problems are termed as Non Deterministic Polynomial. The problems which fall in the class NP may or May not have Polynomial time Running algorithms and therefore are termed as **N.P Problems**

NP Completeness Cont...

Deterministic Vs Non-Deterministic

- A deterministic algorithm is an Algorithm in which there is a define sequence of steps and after each step the algorithm moves to other step and that step is unique.
- A Non Deterministic algorithm has some certain sequence of steps such that after taking one step the algorithm has a number of choices which can be followed and one of these choices can be exercised at random.

NP Completeness Cont...

- The Non Deterministic algorithm takes that step from the choices available which takes it to the solution and only the correct step is taken by the algorithm.
- A fully non-deterministic algorithm cannot be designed in the real world and it is more a theoretical concept.

NP Completeness Cont...

Why are we concerned?

- NP completeness is a theoretical construct But is use full so that a programmer may not unnecessarily struggle to obtain a polynomial time Running algorithm for a problem which actually an NP complete Problem.
- If a problem is marked as NP complete then a programmer may switched to one of the following steps rather then wasting his time in obtaining a polynomial time running algorithms (Which is Non-Deterministic).

NP Completeness Cont...

Various Steps to deal with such algorithms

Step1: To make use of any approximation algorithm and settle down for a polynomial time algorithm to reach a near optimal solution.

Step2: To obtain a better abstraction of the problem. There may be aspects of the problem which may have been ignored during the abstraction process and which may have rendered NP nature to the problem, the problem needs to be abstracted again.

Step3: To Learn the programming basics again.

NP Completeness Cont...

Reducibility

- A problem Q can be reduced to another problem Q’, any instance of it produces a solution which is same as the solution of Q, then Q is said to reducible to Q’.

  Example:-

  Q: ax+b, can be transformed as

  Q’:$0.x^2+ax+b$

  Solution of Q’ is same as that of Q
- Thus we can generalize this concept as a language L1 is polynomial time reducible to a language L2.

NP Completeness Cont...

Polynomial time Verification

- There are certain Problem which can be solved in linear time and there are certain problem which can be solved in polynomial time.
- Hamiltonian cycles: The problem of finding a Hamiltonian cycle in an undirected graph has been studied for over a hundred years. A Hamiltonian cycle of an undirected graph G=(V,E) is a simple cycle that contains each vertex in V. A graph that contains a Hamiltonian cycle is said to be Hamiltonian.
- Verification:-
- Given a graph G is Hamiltonian and then offer to prove it by giving you the vertices in order along the Hamiltonian cycle.

NP Completeness Cont...

- It would certainly be easy enough to verify the proof: Simply verify that the provided cycle is Hamiltonian by checking Whether it is a permutation of the vertices of V and whether each of the consecutive edges along the cycle actually exists in the graph. This verification algorithm can certainly be implemented to run in polynomial time. Thus, a proof that a Hamiltonian cycle exits a graph can be verified in polynomial time.

NP Completeness Cont...

- A vertex cover for G is a set of vertices that covers all the edges in E.
- In this problem we have to find a vertex cover of minimum size in a Given graph.
- Proof:

  Let G=(V,G) is a graph.

  Let K=0

  Then Vertex cover V' subset of V.

  Now k= |V'|.

  For all (u,v)?E

  Now u? V' , v? V'

NP Completeness Cont...

- This verification must be performed in polynomial time.
- But in polynomial time verification doesn't provide the optimal solution But the solution is near to optimal that is why this problem is considered as NP Complete.

 U4. 13

NP Completeness Cont...

Following problems are NP-Complete.

- Vertex Cover Problem.
- TSP (Traveling Sales Man Problem).
- Hamiltonian Cycle Problem.
- Clique Problem.
- Subset Sum Problem.

 U4. 14

SAT

- **Satisfiability** (often written in all capitals or abbreviated **SAT**) is the problem of determining if the variables of a given Boolean formula can be assigned in such a way as to make the formula evaluate to TRUE.
- Equally important is to determine whether no such assignments exist, which would imply that the function expressed by the formula is identically FALSE for all possible variable assignments.

Cook's theorem: Cook's Prove that

SAT is NPC .

 U4. 15

Backtracking

- Suppose you have to make a series of *decisions*, among various *choices*, where
  - You don't have enough information to know what to choose
  - Each decision leads to a new set of choices
  - Some sequence of choices (possibly more than one) may be a solution to your problem
- Backtracking is a methodical way of trying out various sequences of decisions, until you find one that "works"

N-Queen Problem

- A classical problem.
- *n* x *n* chess board
- *n* queens (even number) on the same board
- Queen attacks other at the same row, column or diagonal line
- ➔No 2 queens attack each other

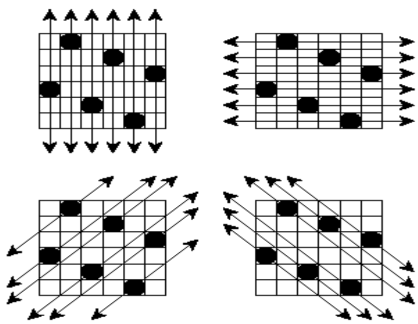
N-Queen Problem Cont...

For example N=6.

- 6 x 6 chess board
- 6 queens on the same board
- Queen attacks other at the same row, column or diagonal line
- ➔No 2 queens attack each other

Now arrange the 6 Queens on the 6 x 6 chess board.

N-Queen Problem Cont...



 U4. 19

Introduction to Computability

Introduction to Computability

- Computability theory is a branch of mathematical logic that originated in the 1930s with the study of computable functions.
- computable function, mathematicians often used the term effectively calculable.

 U4. 20

Introduction to Computability Cont...

- This term has since come to be identified with the computable functions. Note that the effective computability of these functions does not imply that they can be efficiently computed. (i.e. computed within a reasonable amount of time).

 U4. 21

Branch and bound

- An algorithmic technique to find the optimal solution by keeping the best solution found so far.
- For instance, suppose we want to find the shortest route from Zarahemla to Manti, and at some time the shortest route found until that time is 387 kilometers.

Branch and bound Cont...

- According to this techniques,
- In the first iteration this give the solution 387 kilometers.
- After that , this technique will analysis further possible options and found the new result and compare the first result with the newly available result, then it will return the best among them.
- This process will continue till no further options are available to evaluate.

Conclusion

- The special class of problems which include such problems for which algorithms with polynomial Running times have not be designed these problems are termed as Non Deterministic Polynomial.
- NP completeness is a theoretical construct But is use full so that a programmer may not unnecessarily struggle to obtain a polynomial time Running algorithm for a problem which actually an NP complete Problem.

Review Questions

1. Vertex Cut is also known as ___________
2. __________ algorithm always take a shift of one element in case of mismatch.
3. Articulation Point is also known as __________
4. A problem Q can be reduced to another problem Q this phenomenon is known as _____________
5. For Some Algorithms Polynomial time verification doesn't provide the optimal solution But the solution is _____________ that is why this problem is considered as NP Complete.

Review Questions Cont...

6.Consider three decision problems P_1, P_2 and P_3. It is known that P_1 is decidable and P_2 is undecidable. Which one of the following is TRUE?

a. P_3 is decidable if P_1 is reducible to P_3

b.P_3 is undecidable if P_3 is reducible to P_2

c.P_3 is undecidable if P_2 is reducible to P_3

d.P_3 is decidable if P_3 is reducible to P_2's complement

Review Questions Cont...

7.Consider three problems P_1, P_2 and P_3. It is known that P_1 has polynomial time solution and P_2 is NP-complete and P_3 is in NP. Which one of the following is true. P_3 has polynomial time solution if

a.P_1 is polynomial time reducible to P_3

b.P_3 is NP complete if P_3 is polynomial time reducible to P_2

c.P_3 is NP complete if P_2 is reducible to P_3

d.P_3 has polynomial time complexity and P_3 is reducible to P_2

Review Questions Cont...

8.The number of simple digraphs with |V|= 3 is
(a) 29(b) 28(c) 27(d) 26(e) 25

9.The number of simple digraphs with |V|= 3 and exactly 3 edges is
(a) 92 (b) 88 (c) 80 (d) 84 (e) 76

10The number of oriented simple graphs with |V|= 3 is
(a) 27 (b) 24 (c) 21 (d) 18 (e) 15

Review Questions Cont...

1. Explain NP Hard Problems
2. Write the procedure for Matching String Using Naïve Algorithms
3. Explain Reducibility with the help of an example
4. Explain Rabin-Karp Algorithm with the help of an Example
5. How string matching can be implemented through finite automata? Explain with the help of an example.

Review Questions Cont...

6. State and prove cook's theorem.
7. Discuss with the help of an example Assignment problem in detail.
8. Explain NP -Hard?
9. Explain Vertex Cover problem?
10. Explain Subset Sum problem?

Review Questions Cont...

1. Discuss NP-Completeness? List all the NP-Complete Problem and explain any two.
2. Discuss R.Karp Algorithm of String Matching with the help of an example? While solving the example take value of q=3.
3. With reference to NP Completeness: How TSP problem is related with HCP. Explain with the help of example.
4. What do you mean by Polynomial time verification? How this phenomenon is related with various NP-Complete Problems?
5. Explain KMP Algorithm with the help of an Example.

Review Questions Cont...

6. Q5. With reference to NP Completeness: How TSP problem is related with HCP. Explain with the help of example
7. Prove that Vertex Cover problem is NPC.
8. State and prove cook's theorem for SAT.
9. Discuss with the help of an example Assignment problem in detail.
10. What do you mean by n-Queen's Problem?

Suggested Reading/References

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein, “Introduction to Algorithms”, 2nd Ed., PHI, 2004.
2. A. V. Aho, J. E. Hopcroft, J. D. Ullman, “The Design and Analysis of Computer Algorithms”, Addition Wesley, 1998.
3. Ellis Horowitz and Sartaz Sahani, “Computer Algorithms”, Galgotia Publications,1999.
4. D. E. Knuth, “The Art of Computer Programming”, 2nd Ed., Addison Wesley, 1998

