07.09.20

Tutorial

JOVIAL JOE
1ES17CS016

Q1.) What is polynomial time reducability?

Q2.) Prove that Ham Cycle - is NP complete.

Aswers

A1.) ~ A polynomial-time reduction is a method for solving one problem using another.

~ The essence of reduction lies here :

- The main consequence of "A reduces to B" is "if we can solve B then we can solve A". In a sense that tells us that "A is at most as hard a B is"

~ Polynomial reduction is a special type of reduction in which the reduction step could be carried out in polynomial time.

~ A consequence of the reduction step being carried in polynomial time is that, if we can solve B in polynomial time then one can also solve A in polynomial time.

- Polynomial time reduction affirms that if no efficient algorithm exists for the first problem (A), then none exists for the seond either.

- The three most common types of poly-nomial time reductions are

(i) Turing Redution (or cook reductions)

- A polynomial time turing reduction form a problem A to problemB is an algorithm that solves problem
- A using polynomial number of calls to a subroutine, for porob lem B.

(ii) Many-One Reductions (Karp reduations or polynomial transforms)

- A polgnomial-time many one redution from a proben A to probem B is an algorithm for transforming ~~input to~~ to problem A, into input to problem B.

(iii) Truth-table Reductions

- It is a algorithm ~~problem from~~ transforming input to problem A, into inputs to problemB, such that the output for the original problem can be expressess as a function of outputs for B.

A 2.)

- First P is not opposite of NP. P is polynomial time, and NP is non-deterministic polynomial time.
- A language is NP if a proposed solution can be verified in polynomial time.
- NP-complete means most difficult NP problems
  ⇒ if we can solve an NP-complete problem efficiently we can solve all NP problems efficiently.
- In order to prove NP-completness we first show it belongs to NP, by taking a certificate. This certificate is a set of N vertices making up the Hamiltonian cycle.
- To check if this list (of vertices) is actually a solution to the Hamiltonian cycle problem, one counts the vertices to make sure they are all there.
- Then it checks that each is connected to the next by an edge, and that the last is connected to the first.
- This is the so-called verification algorithm.
- This algorithm takes time proportional to n, because, there are n-vertices to count and n-edges to check.

~ n is a polynomial so the check runs in polynomal time. O(n) time.

~ Thereforre the Hamiltonan cycle is ~~NP~~. in NP.

~ To show that it is in NP- complete we have to prove that every other problem in NP can be polynomial - time reduced to Hamiltonian Cycle.

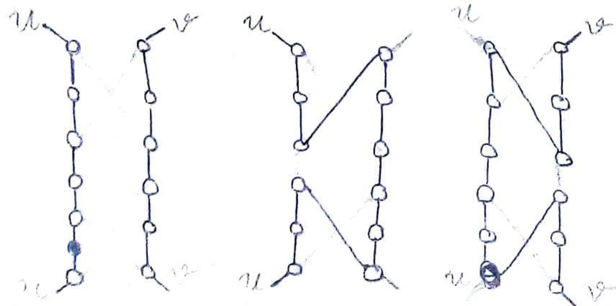
~ Every problem in NP can be polynomial - time reduced to any NP- complete problems.
(eg:- Vertex Cover to Hamiltonian Circuit).

Reduction : Vertex Cover to Hamiltonian Cycle.

Definition: Vertex cover is a set of vertices that touches all edges in the grap.

- Given a grap G and integer k, construct a graph G' ~~and~~ such that G has vertex cover size k iff G' has a Hamilonian cycle.

Idea : To construct a widget for each edge in the graph.



ie. u, v in the graph G, creates a widge t as shown below.

~ As shown above, there are three way to traverse a widget.

(i) Enter from u, go somewhere else in the graph then comback through the other side i.e v.

(ii) Enter & exit through u

(iii) Enter an exith through v.

~ Constructs G' for G (vertex cover) of size k = 2

~ Whith the constrution, any graph with a vertex cover, can be used to mak a graph with a Hamiltonian Cycle graph.

~ Since creating such a graph can be done under polynomial time, simply replace edge with widges and make proper conect and we would have a reduction from vertex cover to hamiltonian cycle.

~ Therefore Hamiltonian Cycle is an NP- complete problem.