Introduction to NP

Instructor: Neelima Gupta

http://people.du.ac.in/~ngupta/

Table of Contents

Problems to be discussed

Clique

Vertex Cover

Hamiltonian Cycle

Independent Set

Set Cover

Subset Sum

Optimization Problems we have seen

- Weighted Interval Scheduling
- Matrix Chain Multiplication
- Segmented Least Squares
- Knapsack
- Sequence Alignment
- Shortest Path

– Solvable in Polynomial Time by Dynamic Programming or Greedy Approach except 0-1 Knapsack which is Pseudo-polynomial.

Optimization Vs Decision Problems

In case of **Optimization** problems, each feasible solution has an associated value, and we want the feasible solution with the 'best' value.

Eg: Shortest path problem

Given an undirected graph (G,s,t), we want to compute the shortest path from vertex s to vertex t (path using fewest edges).

Optimization & Decision Problems

- In case of **Decision** problems, the answer to the problem is a simple 'yes' or a 'no' (more formally, 1 or 0).
- We can obtain a related decision problem for a given optimization problem by bounding the value to be optimized.
- **Eg:** Shortest path problem (Decision Version 1)

  Given an undirected graph (G,s,t,k), is there a path from vertex s to vertex t consisting of exactly k edges?
- Shortest path problem (decision version 2)

  Given an undirected graph (G,s,t,k), is there a path from vertex s to vertex t consisting of at most k edges?

Exercise 1

Give the decision versions (of type 2) for the following problems :

i. Longest Common Subsequence
ii. Matrix Chain Multiplication
iii. Optimal Binary Search Tree
iv. Activity Selection Problem
v. Minimum Spanning Tree

Class NP

- The class of decision problems for which there is a polynomially bounded nondeterministic algorithm.

Or Equivalently

- The class of decision problems which can be verified in polynomial time.

Verification

- A **Verification** algorithm takes as an input, a problem instance, and a certificate and decides whether it is a yes-instance.
- A(x,y) = 1; iff, y is a valid certificate.

x: input instance

y: certificate

Verification: Shortest Path Example

- Let L_1– SP = {<G,s,t,k>: ∋ a path from s to t of length = k}
- Let x ∈ SP, then we claim that ∋ a certificate y such that, <G,s,t,k,y> can be verified in polynomial time.

  - What is that?

    Ans: Since x ∈ SP, ∋ a path (sequence of vertices) from s to t of length = k. This path itself serves as the certificate y.

  - What does the verification algorithm do?
    - Ans: Just check that the given sequence of vertices in y indeed forms a path in G (i.e. check that between every consecutive pair of vertices there is an edge in G) and,
    - That its length is k.

Verification – Running Time

- Adjacency Matrix Representation of Graphs: Checking an edge takes constant time.
- Linked List representation of Graphs:
  - To check an edge, we can scan the adjacency list of one of its end points. Since length of the adjacency list is O(V), it can be checked in O(V) time.

In either case checking an edge takes O(V) time.

How many checks? Ans: O(V)

Thus the total time is $O(V^2)$ time.

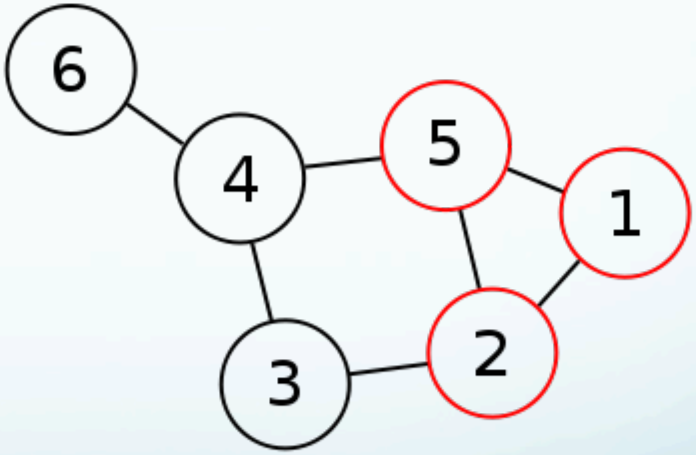
- Hence, we can verify the certificate in polynomial time.

More NP Problems

- Clique
- Vertex Cover
- Hamiltonian Cycle
- Independent Set
- Set Cover
- Subset Sum

Definition: Clique

Given an undirected graph G =(V,E), a clique is a subset V'of V such that every pair of vertices in V' is connected by an edge in E.



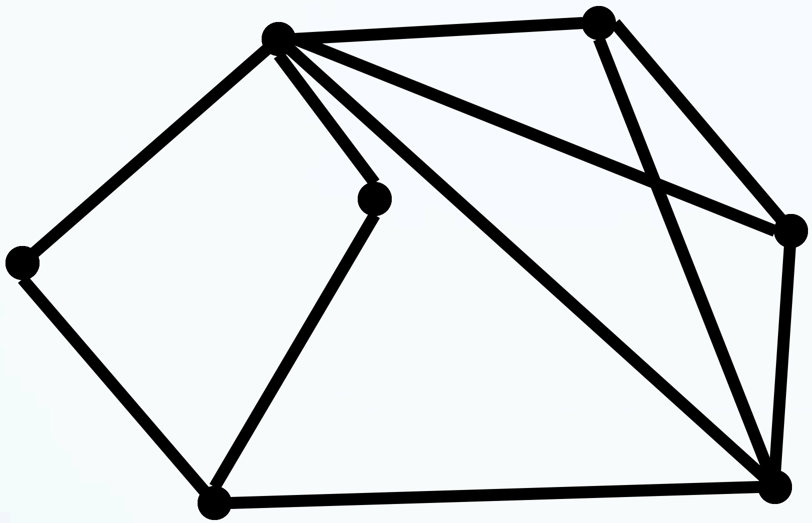
In the graph to the right,

vertices {1,2,5} form a

clique since there is an edge

between every pair of edges of this set.

G



Green ovals represent CLIQUE for this graph

Clique Problem

OPTimization Problem : Given an undirected graph G find a largest Clique in G.

Decision problem: Given G, does there exist a clique of size equal to k in G?

Clique = {(G,k): G contains a clique of size k}

Clique is in NP

- Clique = {<G,k>: G has a clique of size k}
- For x = <G,k> in Clique, does there exist a certificate y: |y| = polynomial in the length of x and a polynomial time algorithm that can use y to verify that x is in Clique?

  - Show the existence of y,
  - Show that |y| = polynomial in the length of |x|,
  - Give an algorithm that verifies x using y,
  - Show that the algorithm runs in polynomial time in |y| and |x| and hence in polynomial time in the length of |x|.

SO, FOUR STEPS TO SHOW THAT A PROBLEM IS IN NP

Verification

- Let x (=(G,k)) ∈ Clique, then we claim that ∋ a certificate y such that, <x,y> i.e. <G,k,y> can be verified in polynomial time.

  - Since x ∈ Clique, ∋ a clique of size k in G (by the definition of the set). This set of vertices itself serves as the certificate y.

  - Clearly |y| is no more than the number of vertices in G. i.e. |y| = O(|x|)

  - What does the verification algorithm do?
  - Ans: Just check that the given set of vertices in y indeed forms a clique in G and,
  - That its size is k.

Verification Algorithm

- Algo: For every pair of vertices u and v in y, check that there exists an edge (u,v) in G

- Running Time: : ${}^{k}C_2$ * degree(u)

$$= {}^{k}C_2 . O(V) = O(k^2).O(V)$$
$$= O(V^2.V)$$
$$= O(V^3)$$

- Checking the |y| is k takes constant time.

- Hence, we can verify the certificate in polynomial time.

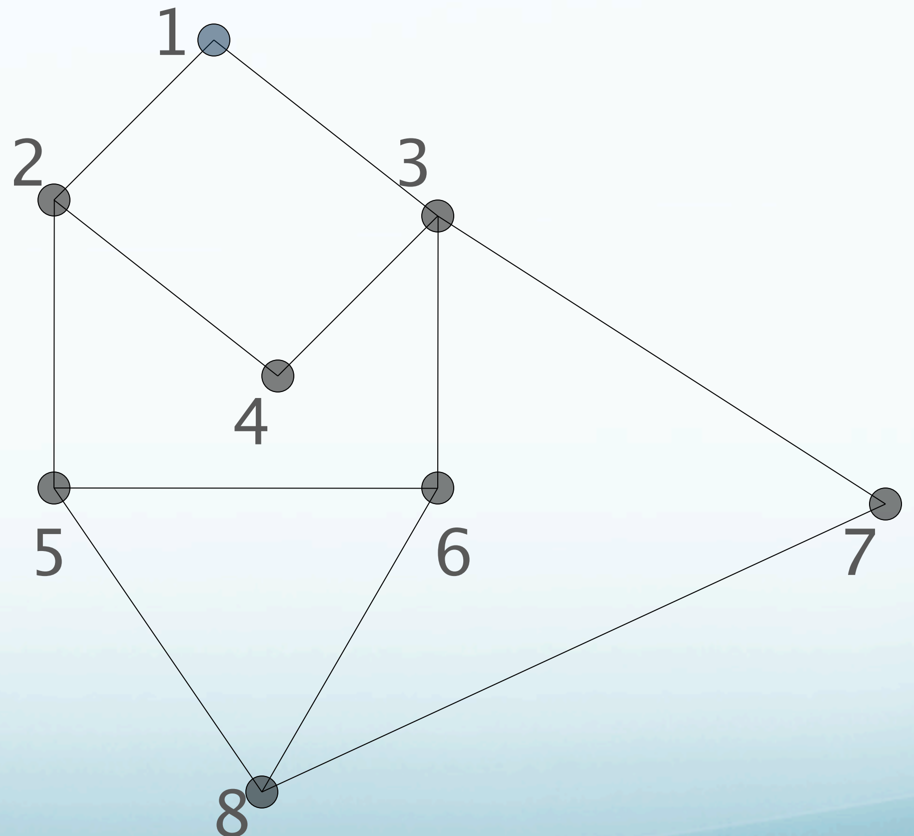
Vertex Cover

- A vertex cover of an undirected graph $G = (V,E)$ is a subset V' of V, such that if an edge $(u,v) \in E$, then at least one of u and v is in V'.

with V` = {1,2,3,4} edges covered are,

with V` = {1,2,3,4,6} edges covered are,

Minimal set V` to cover all edges is {2,3,6,8},



Vertex Cover Problem

OPTimization Problem : Given an undirected graph G find a smallest vertex cover in G.

VC(G,k) : Given G, does there exist a vertex cover of size equal to k in G?

VC = {(G,k): G contains a vertex cover of size k}

Vertex Cover is in NP

- VC = {<G,k>: G has a vertex cover of size k}
- For x = <G,k> inVC, does there exist a certificate y: |y| = polynomial in the length of x and a polynomial time algorithm that can use y to verify that x is in VC?

  - Show the existence of y,
  - Show that |y| = polynomial in the length of |x|,
  - Give an algorithm that verifies x using y,
  - Show that the algorithm runs in polynomial time in |y| and |x| and hence in polynomial time in the length of |x|.

SO, FOUR STEPS TO SHOW THAT A PROBLEM IS IN NP

Verification

- Let x (=(G,k)) ∈ VC, then we claim that ∋ a certificate y such that, <x,y> i.e. <G,k,y> can be verified in polynomial time.
  - Since x ∈ VC, ∋ a vertex cover of size k in G (by the definition of the set). This set of vertices itself serves as the certificate y.
  - Clearly |y| is no more than the number of vertices in G. i.e. |y| = O(|x|)
  - What does the verification algorithm do?
  - Ans: Just check that the given set of vertices in y indeed forms a vertex cover in G and,
  - That its size is k.

Verification Algorithm

- For every edge (u,v) in G check that at least one of the vertices u and v is in y.
- Running Time: $|E|\ |y|$

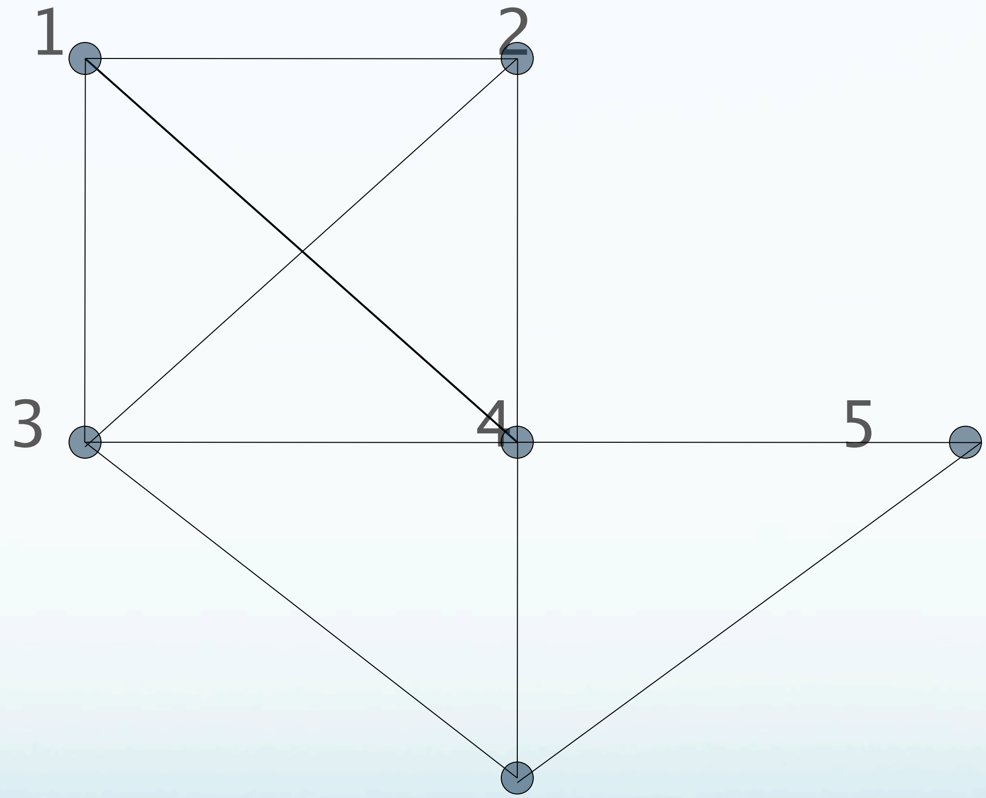
  $=O(E\ V)$

- Checking that |y| is k takes constant time.
- Hence, we can verify the certificate in polynomial time.

Hamiltonian Cycle

- A **Hamiltonian cycle** of an undirected graph, G, is a simple cycle that spans every vertex.

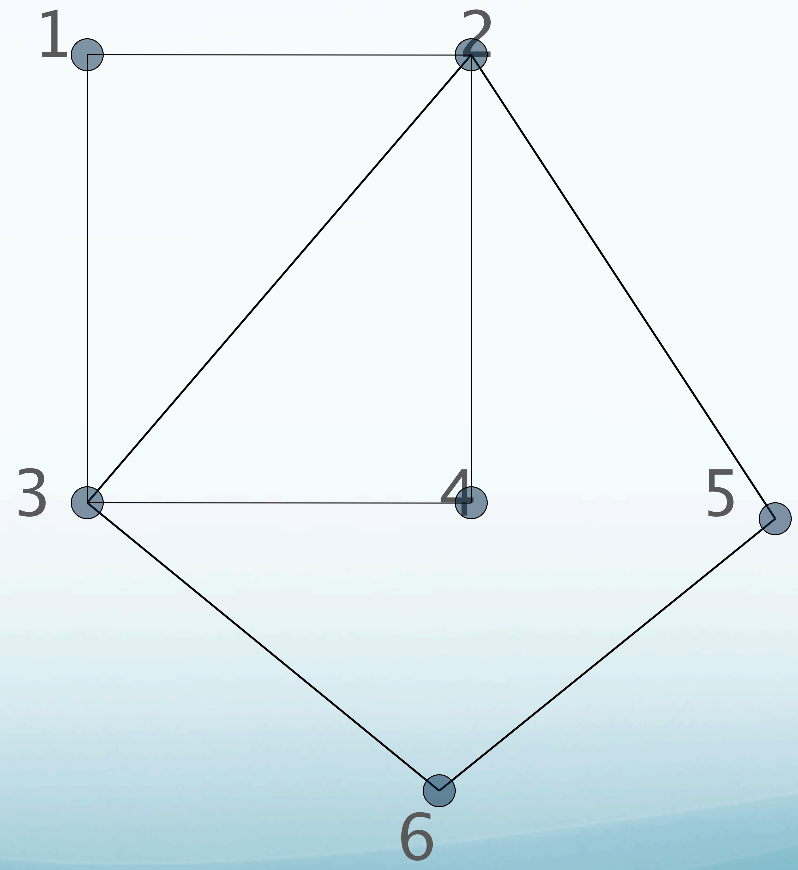
Forming an HC,



- However, for the graph G`, there does not exist any Hamiltonian cycle.

1) with vertices 1,2,3,4 – not an HC

2) with vertices 1,2,5,6,3 – again not an HC



Hamiltonian Cycle

- Problem of HC is to find a HC in the given graph.
- Decision Version: Given an undirected graph G, does G contain a Hamiltonian cycle?
- HC = {(G) : G contains a Hamiltonian cycle}

HC is in NP

- HC = {<G>: G has a Hamiltonian cycle}
- For x = <G> in HC, does there exist a certificate y: |y| = polynomial in the length of x and a polynomial time algorithm that can use y to verify that x is in HC?

  - Show the existence of y,
  - Show that |y| = polynomial in the length of |x|,
  - Give an algorithm that verifies x using y,
  - Show that the algorithm runs in polynomial time in |y| and |x| and hence in polynomial time in the length of |x|.

SO, FOUR STEPS TO SHOW THAT A PROBLEM IS IN NP

Verification

- Let x (=(G) ∈ HC, then we claim that ∃ a certificate y such that, <x,y> i.e. <G,y> can be verified in polynomial time.
  - Since x ∈ HC, G contains a Hamiltonian cycle (by the definition of the set). This sequence of vertices that forms a HC in G itself serves as the certificate y.
  - Clearly |y| is no more than the number of vertices in G. i.e. |y| = O(|x|)
  - What does the verification algorithm do?
  - Ans: Just check that the given sequence of vertices in y indeed forms a Hamiltonian cycle in G.

Verification Algorithm

- Let $y = <u_1, u_2, \ldots u_m>$
- Algo:
  - For i =1 to m-1

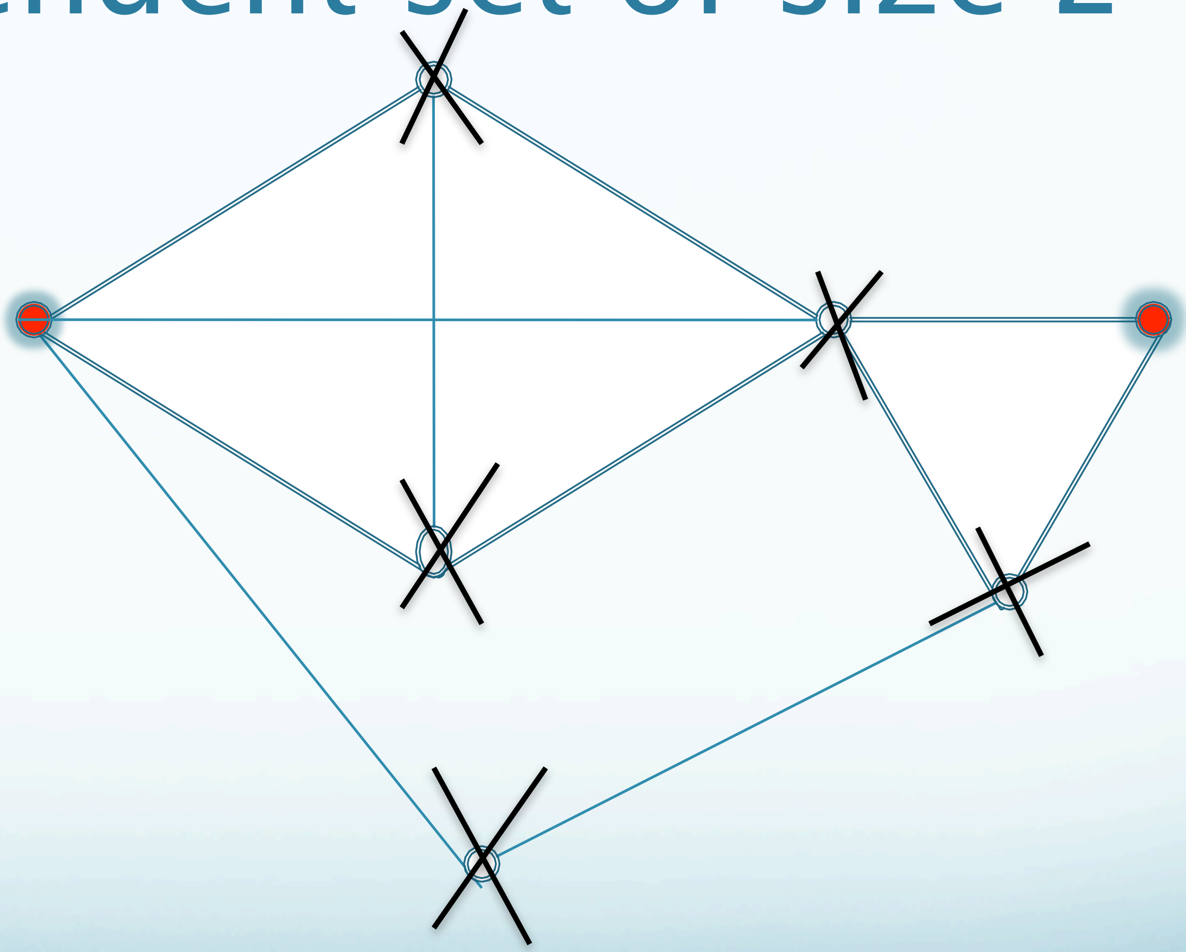
    check that (u_i, u_{i+1}) is an edge in G
  - check that (u_m, u_1) is an edge in G
  - Check that no vertex is repeated.
  - Check that m = n (i.e. all the vertices have been spanned)
- Running Time: $m\,|V| + m^2 + \text{constant}$

  $= O(V^2)$
- Hence, we can verify the certificate in polynomial time.

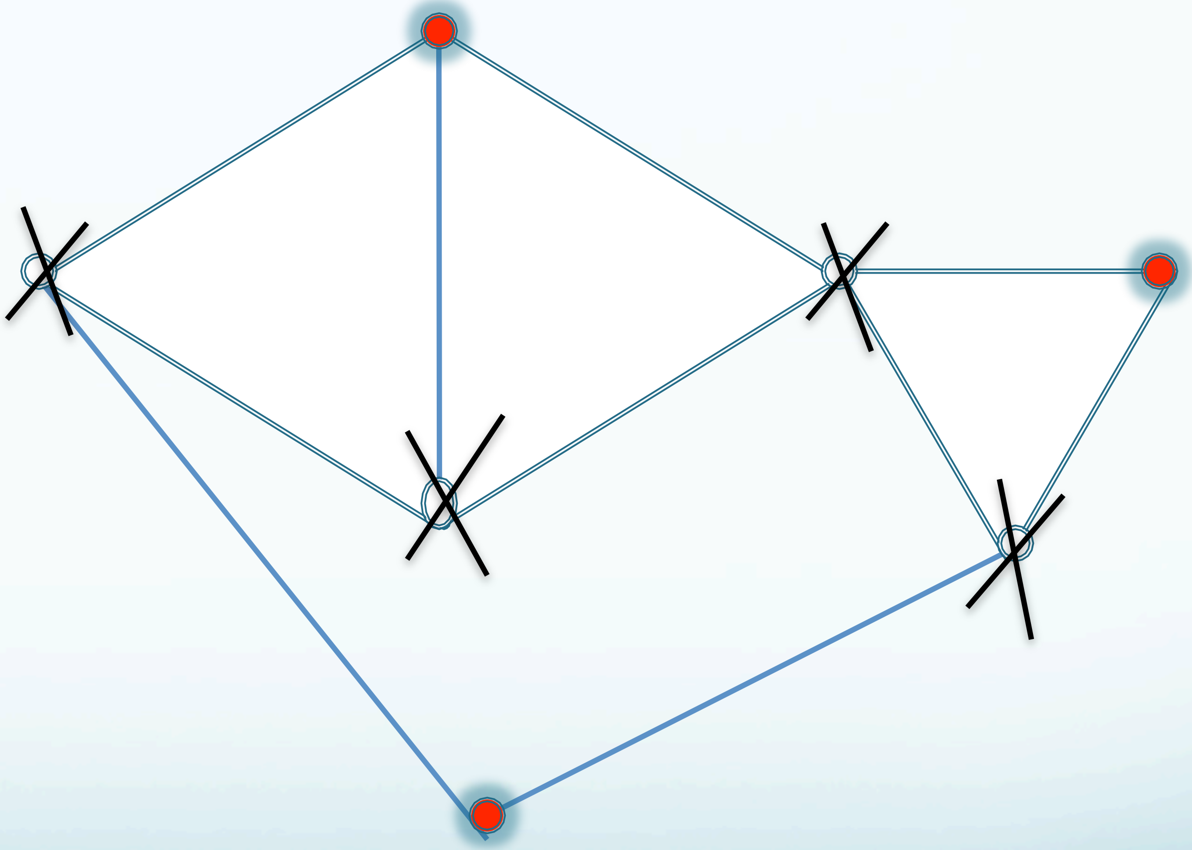
Independent Set

Given a graph G = (V,E), a subset S of V is said to be independent if no two nodes in S are joined by an edge in E.

Independent set of size 2



Independent set of size 3



Independent Set Problem

OPTimization Problem : Given an undirected graph G find a largest independent set in G.

Decision problem: Given G, does there exist an independent set of size equal to k in G?

IS = {(G,k): G contains an independent set of size k}

IS is in NP (Exercise)

- IS = {<G,k>: G has an IS of size k}
- For x = <G,k> in IS, does there exist a certificate y: |y| = polynomial in the length of x and a polynomial time algorithm that can use y to verify that x is in IS?

  - Show the existence of y,
  - Show that |y| = polynomial in the length of |x|,
  - Give an algorithm that verifies x using y,
  - Show that the algorithm runs in polynomial time in |y| and |x| and hence in polynomial time in the length of |x|.

SO, FOUR STEPS TO SHOW THAT A PROBLEM IS IN NP

Set Cover Problem

- Given a collection U of elements and a family S of subsets of U, a cover C is a sub-family of S (i.e. a collection of subsets from S) whose union is U.
- Example: U = {1,2,3,4}
- S1 = {1,2}, S2 = {2,3}, S3 = {1,3,4}, S4 = {3,4}, S5 = {3}, S6 = {4}
- Then S1, S2, S3 is a cover of size 3;
- S1, S2, S6 is a cover of size 3;
- And S3, S1 is also a cover of size 2;
- We are interested in finding a cover of minimum size.

Set Cover is in NP (Exercise)

Subset Sum Problem

Given: A finite set S of integers and a target t ϵ I.

To Find: If there exists a subset S' of S whose elements sum up to t.

Subset Sum is in NP (Exercise)

Acknowledgement

- Scribe by MSc (Computer Science) 2009.
  - Prashant Singh (NP Class)
  - Sapna Grover (P and NP , Vertex Cover, HC….)
  - Soniya Verma (Independent Set)
  - Shivam Sharma (Clique)

End