34.5-1

子图同构问题取两个无向图 G_1 和 G_2，要回答 G_1 是否与 G_2 的一个子图同构这一问题。证明：子图同构问题是 NP 完全的。

Step 1 of 6

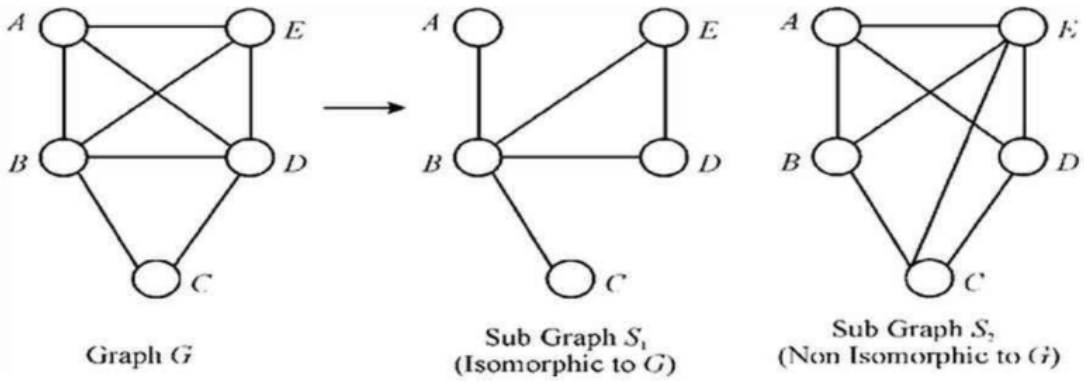
SUB GRAPH-ISOMORPHISM PROBLEM

The sub-graph isomorphism problem represents a computational task which deals with two graphs *G* and *S* provided as an input. This problem decides and determines that whether there exists any sub graph of *S* which is isomorphic to *G*. As like the Maximum Clique Problem and Hamiltonian cycle, the sub graph isomorphism problem is NP complete.

It also plays an important role in case of matching the patterns, comparing tree structures and performing searching operations on strings.

Step 2 of 6

For Example: Consider the graphs given below:



Graph G | Sub Graph S_1 (Isomorphic to G) | Sub Graph S_2 (Non Isomorphic to G)

Step 3 of 6

In the graphs as shown above, it can be seen that sub graph S_1 of graph *S* is isomorphic to graph *G* and the sub graph S_2 is not.

Step 4 of 6

Given a sub graph-isomorphism problem that takes two undirected graphs G_1 and G_2, it is asked that weather G_1 is isomorphic to a sub graph of G_2 and it is also required to show that this problem is NP-complete.

Sub Graph-isomorphism problem is NP complete if it belongs to NP and also it is polynomial reducible to SI, that is, $\text{CLIQUE} \leq_p \text{SI}$. This problem can be considered as NP because it can be verified or checked that the mapping of the vertices of the graph G_1 and the vertices G_2 can be done in the polynomial time in order to deduce that whether Graph G_1 is isomorphic to sub graph of G_1.

Thus, it shows that there exists a non deterministic polynomial time algorithm that can recognize or determine the elements or mapping between the two input graphs and therefore it can be stated that $\text{Sub-Graph isomorphism} \in \text{NP}$.

Also, the fact that Sub Graph- isomorphism problem belongs to the class of NP-hard problems can be proved or shown by considering a well known CLIQUE problem which is already a NP-complete problem.

Thus, in order to show that the Sub Graph-isomorphism problem is NP-hard, an instance from a CLIQUE problem is reduced to an instance of graph G_1 which is isomorphic to sub graph of G_2.

Step 5 of 6

CLIQUE: Basically, a clique is a concept in graph theory that represents a complete sub graph of an undirected graph *G*. It is a subset of vertices which are connected to each other by an edge. Given a graph *G* and an integer *k*, the Clique problem finds that whether a clique of size *k* is in the graph *G*. The procedure for reducing is considered below as:

Step1. Consider the graph $G = G_2$

Step2. And let G_1 be any connected graph with number of vertices *k*.

Step3. Then, G_1 and G_2 are supplied as input into the algorithm or procedure that solves Sub graph isomorphism.

By referring to the above condition, it can be concluded that **Step1** takes $O(|G|)$ time. **Step 2** takes $O(|G|+|G|^2)$ time and **Step 3** takes $O(1)$ time.

Step 6 of 6

Now, going through the step, the time complexity for the algorithm is obtained in polynomial time. And, here is the proof of correctness:

1. If *G*, *k* results to be "yes" or we can say positive instance of the problem, then *G* has a sub graph which is isomorphic in nature to a connected graph whose size is *k* and the SI (sub graph-isomorphism) algorithm answers is in the form of positive that is "yes" or positively to the input G_1 , G_2 .

Hence, in this case, a positive response comes that is "yes" in the CLIQUE algorithm to input *G*, *k*.

2. If *G*, *k* results to be "no" or we can say a negative instance of the problem, then *G* does not have a sub graph means isomorphic to a connected graph having size *k* and answer provided by SI will be "no" or negatively. Hence, in this case the CLIQUE answers "no".

Therefore, if G_1 is isomorphic to a sub graph of G_2 only in such a condition that *G* contains a clique of size *k*. Now, as both sets of the nodes of the graphs are completely connected so G_1 is sub graph isomorphic to G_2 and hence, it has been deduced or proved that this problem is NP-hard.

Thus, $\text{Sub-Graph isomorphism} \in \text{NP}$ is proved and also it is NP hard as CLIQUE is polynomial time reducible to SI. Now, since CLIQUE is in NP-complete, sub graph-isomorphism (SI) is also in NP-complete.

34.5-2

给定一个 $m\times n$ 的整数矩阵 A 和一个整型的 m 维向量 b，0-1 整数规划问题即研究是否有一个整型的 n 维向量 x，其元素取自集合 $\{0, 1\}$，满足 $Ax \leqslant b$。证明：0-1 整数规划问题是 NP 完全的。（提示：由 3-CNF-SAT 问题进行归约。）

Step 1 of 5

Step 1:- IntProg $=\{ < A,b >: A$ is an $n \times m$ integer matrix, b is an m – vector of integers and there exists 0 vector $x \in \{0,1\}^n$ such that $Ax \geq b\}$. The Idea for the reduction: 3-CNF – SAT $\leq_p$ IntProg

Ex: $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee \neg x_4)$

Transform each clause into an inequality, transforming each literal into expression x_i for $(1 - x_i)$ depending whether it is a variable or its negation that appears in the clause.

$(x_1 \vee x_2 \vee x_3): x_1 + x_2 + x_3 \geq 1$

$(x_1 \vee \neg x_3 \vee \neg x_4): x_1 + (1 - x_3) + (1 - x_4) \geq 1$

Which is equalent to $x_1 - x_3 - x_4 \geq -1$ So, the corresponding instance of IntProg is $A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & -1 & -1 \end{bmatrix}$ $b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Step 2 of 5

Step 2: Reduction Algorithm

Algorithm $F(<\phi>)$

Check whether ϕ is in 3–CNF format if it is not, then return $(<A=[1], b=[2]>)$; let m be the number of clauses and n be the number of variables in ϕ

```
1.  for i = 1 to m do
2.  NUMNEGATED ← 0;
3.  for j = 1 to m do A[i, j] = 0;
4.  for each literal L in C_i do;
5.     if  L = x_j then
6.        A[i, j] = a[i, j]+1;
7.     Else if 1 = 7x_j then
8.        A[i, j] = a[i, j]−1;
9.        NUMNEGATED ++;
10.    End if
11. End for
12.. b[i] = 1 − NUMNEGATED
13. End for
14. Return (<A,b >)
```

Step 3 of 5

Step 3: $<\phi> \in -CNF-SAT \Leftrightarrow <A, b> \in$ IntProg:

If ϕ is not in 3 – CNF format then $A=[1]$ and $A=[2]$, and the system $1x_1 \geq 2$ has no solution for $x_1 \in 0,1$. It remains to look at the case that ϕ is in 3 – CNF format. In this case, ϕ is satisfiable if and only if there exists a truth assignment to $x_1, x_2 \cdots\cdots x_n$ such that this is true if and only if there exits value 0, 1 assigned to variables $x_1, x_2 \cdots\cdots x_n$ such that $A_x > b$. Let us analyze each clause C_i. C_i is satisfied by $x_1, x_2 \cdots\cdots x_n$ if and only if at least one of its literals is assigned the value true. Let y_1, y_2, y_3 correspond to the truth values of each literal in C_i. Thus C_i is satisfied if and only if at least one of y_1, y_2, y_3 is equal to 1, which is equalent to saying that $y_1 + y_2 + y_3 \geq 1$.

Step 4 of 5

Now we relate be values of y_1, y_2, y_3 with the values of $x_1, x_2, \cdots x_n$. Let l_1, l_2 and l_3 be the literals in C_i. Then if $l_j = x_{ij}$ then $y_i = x_{ij}$, but if $l_j = 7x_{ij}$ then $y_i = 1 - x_{ij}$ since $y_1 = 1$ if $x_{ij} = 0$ and $y_1 = 1$ if $x_{ij} = 1$. Substituting this into the equation $y_1 + y_2 + y_3 \geq 1$ we get $a_{k_1} x_{k_1} + a_{k_2} x_{k_2} + a_{k_3} x_{k_3} \geq 1 - n_i$

$$\text{Where } a_{k_1} = \begin{cases} 1 & \text{if } l_i = x_{ij} \\ -1 & \text{if } l_j = 7x_{k_1} \end{cases}$$

And n_i is the number of variables in C_i that appear negated. This ϕ is satisfiable if and only if there exists a $0-1$ assignment variables $x_1, \cdots\cdots, x_n$ such that $A_x \geq b$.

Step 5 of 5

Step 4: F runs in Polynomial time:

Checking whether ϕ is in $3-$ CNF format can be done in linear time on the number of clauses. The loop on i runs in m steps, The loop on j runs in n steps The loop on the literals run in constant number of steps, since there are only 3 literals per clause. So the running time of F is O(nm).

34.5-3

整数线性规划问题与练习 34.5-2 中给出的 0-1 整数规划十分相似，区别仅在于向量 x 的值可以取任何整数，而不仅是 0 或 1。假定 0-1 整数规划问题是 NP 难度的，证明：整数线性规划问题是 NP 完全的。

Step 1 of 1

The problem can be expressed in the form:

Min. $f = x_1 2^2 x_2 + \ldots\ldots + n^2 x_n$

Subject to $x_1 + 2x_2 + \ldots\ldots + nx_n = p$

$x_1 + x_2 + \ldots\ldots + nx_n = q$

$x_k \geq 0$, integer, $\forall k$,

Where the integer $n \geq 2$ and the parameters p and q are positive integers with $p \geq q > 0$

Applying Cramer's rule from linear algebra to the constraint.

$$x_1 = \frac{p - jq}{i - j}, \quad x_j = \frac{ig - p}{1 - j}, (1 \leq i < j \leq n)$$

34.5-4

证明：如果目标值 t 表示成一元形式，试说明如何在多项式时间内解决子集和问题。

Step 1 of 1

To show that subset-sum problem is solvable in Polynomial time, we need to give verification procedure. Given S and t, the certificate in just the indices of the numbers that form the subset S! we can add two b – bit numbers together in O(b) time. So in polynomial time we can computer sum of elements of S^1, and verify that this sum equals t.

34.5-5

集合划分问题的输入为一个数字集合 S。问题是：这些数字是否能被划分成两个集合 A 和 $\overline{A}=S-A$，使得 $\sum_{x\in A}x=\sum_{x\in\overline{A}}x$ 。证明：集合划分问题是 NP 完全的。

Step 1 of 1

Using the portion of the set as certificate shows that the problem is in NP. For NP-hardness we give reduction from subset sum . Given an instance $S=\{x_1,x_2,\cdots\cdots x_n\}$ and t compute r such the $r+t=(\sum_{x\in S}x+r)/2\ \ (r=\sum_{x\in S}x-2t)$. The instance for set-partioned assume S has a subset S' summing to t. then the set S' U r Sums to r +t and thus partions R. Conversly, if R can be portioned then the set containing the element r sum to r + t. all other eliments in this set sums to t thus providing the subset S'.

34.5-6

证明：哈密顿路径问题是 NP 完全的。

Step 1: HAMPATH ∈ NP

CERTIFICATE: A sequence of vertices y

VERIFICATION: $a(<G,u,v>,<y>)$

1. Check whether y has n vertices; If not return 0;
2. Check whether $y=(y_1,y_2\cdots\cdots,y_n)$ 0; Repeated vertices; if so return 0;
3. Check whether $(y_i,y_{i+1})\in$ E for $i=1,\cdots,n-1$ And whether $\{y_n,y_1\}\in E$. If some of the tests fail then return 0;
4. Check whether $y_1=u$ and $y_n=V$. If not return 0; otherwise return 1;

Step 2 of 8

A runs in Polynomial time Since

1. runs in O(n) steps
2. runs in O(n) steps
3. runs in O(n) steps
4. runs in O(n) steps So A runs in O(n) steps

If is easy to see A is a correct verification algorithm for HAMPATH.

Step 3 of 8

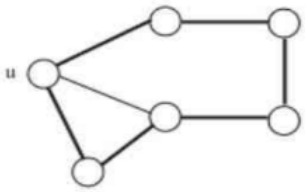
Step 2: HAMCYCLE ≤ p HAMPATH

Idea for the reduction. Given G, pick an arbitrary vertex u and create a new Vertex u' connected it to all the neighbours of u, in the new graph G'. A hamiltonian in G corresponds to a hamiltonian path from u to u' in G'.

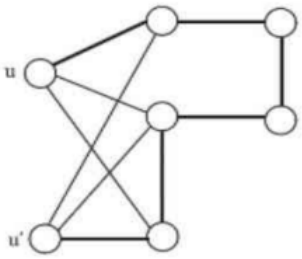
Step 4 of 8

Example:

G:



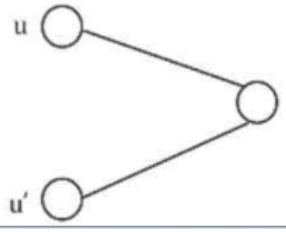
G':



Step 5 of 8

The only case for which this reduction fails is $G: u 0-0$ Since G has no Hamiltonian cycles but has a Hamiltonian path between u and u'

G':



Therefore we treat the case $(V)=2$ separately in the algorithm in the algorithm.

Step 6 of 8

Step 3:- Reduction Algorithm

Algorithm $F(<G>)$

Let $G=(V,E)$

If $|V|=2$ then return $<G'=(V'=\{u,v\},E'=\phi)$ $<G'=(V'=\{u,v\},E'=\phi),\mu,V>$

Select a vertex u in G

$E' \leftarrow E$

For each bv inV do

If $\{u,v\} \in E$ then $E' \leftarrow E' \bigcup \{u'v\}$

Return $<G'=(V',E'),u,u'>$;

Step 7 of 8

Step 4: the reduction works that is G has a Hamiltonian cycle if and only if G' has a hamiltaonian path from u to u' $(\Rightarrow)$ let $C=(V_1=u,V_2,\cdots\cdots V_n)$ be a Hamiltonian cycle in G (we can assume $V_i=u$ with loss of generately) it is easy to see that $(V_1=u,V_2,V_3-----V_n,V_{n+1}=u')$ is a hamiltonain path between u and u' in G' since $(V_1,V_2-------V_n)$ are distinct vertices in G so $(V_1=u,V_2,-----V_n,V_{n+1}=u')$ are distinct vertices in G': more over if $\{V_i,V_{i+1}\}\in E$ $i=1,2\cdots\cdots,n$ and $\{V_n,u\}\in E$ then $\{V_i,V_{i+1}\}\in E'$, $i=1,2,\cdots\cdots n$ and $\{V_n,u'\}\in E'$ $(\Leftarrow)$ Let $P=(u=V_1,V_2,V_3-----,V_n,V_{n+1}=u')$ be a Hamiltonian path in G' then $(V_1,V_2,\cdots\cdots V_n,V_{n+1}=u')$ are distinct vertices in G'. So $(V_1,V_2,------V_n)$ are distinct in G. more over, since $\{V_i,V_{i+1}\}\in E'$, $i=1,2,-------n$ and $\{u',V_1\}\in E'$ then we conclude $\{V_i,V_{i+1}\}\in E$, $i=1,2,\cdots\cdots n$ and $\{u,v_1\}\in E'$.Moreover, since $n>2$, then $\{V_1,V_2\}\neq\{V_n,V_i\}$ So, $(V_1,V_2,--------,V_n)$ is a Hamiltonian cycle in G.

Step 8 of 8

Step 5: F runs in Polynomial time

Copying G into G', takes time $O(n^2)$ where $n=|V|$. Creating u' and its incidence edges takes time in $O(n)$. There fore, F runs in $O(n^2)$.

34.5-7

最长简单回路问题是在一个图中，找出一个具有最大长度的简单回路(无重复的顶点)。

证明：这个问题是 NP 完全的。

Step 1 of 6

Step 1: LONGESTPATH $\in$ NP

1. An algorithm $A(x,y)$ to verify LOGESTPATH is described here
   - the certificate y for the algorithm will be a sequence of vertices in the graph.
   - The verifier will check that the vertex sequence y satisfies the following conditions.

- y has at least K vertices.
- y has no repeated Vertices (since the path must be simple)
- The first Vertex of y is u and the last one is V.
- There is an edge connecting each pair of contiguous vertices in the sequences y

The Verifier returns 1 if and only if all these conditions are satisfied.

Step 2 of 6

2. This algorithm is a Verifier.
   - if $<G,u,v,k> \in$ LONGESTPATH, then the longest simple path corresponds to a sequence of vertices y with no more than n vertices (so the certificate is short). For this y, $A(x,y)=1$ since the longest path length at least k.
   - if $<G,u,v,k> \notin$ LONGESTPATH, then there exists no vertex sequence corresponding to a simple path with length at least K.so for any y, one of the conditions checked by the algorithm be violated, and this $A(x,y)=0$.
3. The algorithm runs in Polynomial time, Since the conditions can easily be checked in Polynomial time.

Step 3 of 6

Step 2: We will show that HAMPATH $\leq$ p LONGESTPATH

Step 4 of 6

Step 3: We will describe a Polynomial time algorithm f that computers a function
$f:\{0,1\}^* \rightarrow \{0,1\}^*$ such that f(< G,u,v >) $\in$ LONGESTPATH
Algorithm F:Input $<G,u,v>$
Let n be the number of vertices in G
Return (G, u, v, n>

Step 5 of 6

Step 4: we now prove that $<G,u,v> \in$ HAMPATH
If and only if $f(G,u,v) = <G,u,v,n> \in$ LONGESTPATH
Note that a Hamiltonian path from u to v is a path passing through every vertex of G exactly once. So, a Hamiltonian path from u to v is a simple path from u to V with n vertices and (Vice Versa).If $<G, u, v,> \in$ HAMPATH then there exists a Hamiltonian path P from u to V with n vertices. So P is a simple u to V with n vertices. So P is a simple path with n vertices which implies $<G, u, v, n> \in$ LONGESTPATH if $<G, u, v> \notin$ HAMPATH then there exists no Hamiltonian path from u to v in G. So every simple path from u to V in g has at most n – 1 vertices. So $<G, u, v, n> \notin$ LONGESTPATH.

Step 6 of 6

Step 5: We show that algorithm F runs in Polynomial time:The algorithm does nothing but reading the input and outputting what was on the input and the number n which is similar than the input size. So the algorithm runs in linear time on the input size.

34.5-8

在半 3-CNF 可满足性中，给定一个 3-CNF 形式的公式 ϕ，它包含 n 个变量和 m 个子句，其中 m 是偶数。我们希望确定是否存在对 ϕ 中变量的一个真值赋值，使得 ϕ 中恰有一半的子句为 0，同时恰有另一半的子句为 1。证明：半 3-CNF 可满足性问题是 NP 完全的。

No answer~