Some Review Problems

Data Structures

Give one advantage of a Linked List, as compared to

an array.

Ans.   Linked List has much faster Insert and Remove operations than an Array – O(1) for Linked List and O(n) for Arrays.

Another one – Linked List allows more flexible memory management than an Array. It does not require large contiguous memory blocks, and needs no preallocation.

Data Structures – Hashing 1

Insert the following sequence of numbers 23, 46, 12, 21, 75, 5, 3

into a hash table of size 9 using h(x) = x%9 as a hash function. % means

Mod. Use Chaining with Linked List to avoid collision.

Ans.

0	1	2	3	4	5	6	7	8
	46		12,21, 75,3		23,5			

Data Structures – Hashing 2

Suppose that we store n keys in a hash table of size $m = n^2$ using a hash function h randomly chosen from a universal class of hash functions. Then, the probability is less than $1/2$ that there are any collisions.

Some Review Problems

Suppose that A and B , and that A ≤p B. Circle one answer for each statement below.

If A ∈NP-Complete and B is in NP then B ∈ NP-Complete.

True   False

If B ∈ P then A ∈ P.

True   False

If B is NP-hard then A is NP-hard.

True   False

Reduction -1

3-CNF to Clique

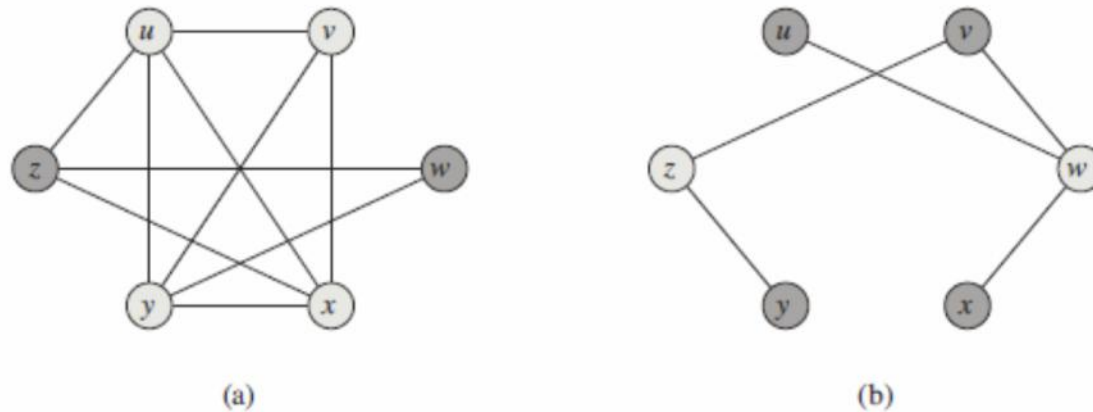To be done in the Class

Clique → Vertex Cover

Figure 34.15 Reducing CLIQUE to VERTEX-COVER. (a) An undirected graph $G = (V, E)$ with clique $V' = \{u, v, x, y\}$. (b) The graph $\overline{G}$ produced by the reduction algorithm that has vertex cover $V - V' = \{w, z\}$.

Clique → Vertex Cover: **Problem**

Suppose that G has a clique $V' \subseteq V$ with $|V'| = k$. We claim that $V - V'$ is a vertex cover in $\overline{G}$. Let (u, v) be any edge in $\overline{E}$. Then, $(u, v) \notin E$, which implies that at least one of u or v does not belong to V', since every pair of vertices in V' is connected by an edge of E. Equivalently, at least one of u or v is in $V - V'$, which means that edge (u, v) is covered by $V - V'$. Since (u, v) was chosen arbitrarily from $\overline{E}$, every edge of $\overline{E}$ is covered by a vertex in $V - V'$. Hence, the set $V - V'$, which has size $|V| - k$, forms a vertex cover for $\overline{G}$.
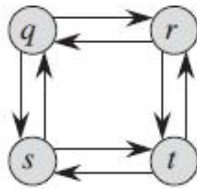
Clique → Vertex Cover: Solution

Suppose that G has a clique $V' \subseteq V$ with $|V'| = k$. We claim that $V - V'$ is a vertex cover in $\overline{G}$. Let (u, v) be any edge in $\overline{E}$. Then, $(u, v) \notin E$, which implies that at least one of u or v does not belong to V', since every pair of vertices in V' is connected by an edge of E. Equivalently, at least one of u or v is in $V - V'$, which means that edge (u, v) is covered by $V - V'$. Since (u, v) was chosen arbitrarily from $\overline{E}$, every edge of $\overline{E}$ is covered by a vertex in $V - V'$. Hence, the set $V - V'$, which has size $|V| - k$, forms a vertex cover for $\overline{G}$.

Clique → Vertex Cover: Soln. - **Converse**

Conversely, suppose that $\overline{G}$ has a vertex cover $V' \subseteq V$, where $|V'| = |V| - k$. Then, for all $u, v \in V$, if $(u, v) \in \overline{E}$, then $u \in V'$ or $v \in V'$ or both. The contrapositive of this implication is that for all $u, v \in V$, if $u \notin V'$ and $v \notin V'$, then $(u, v) \in E$. In other words, $V - V'$ is a clique, and it has size $|V| - |V'| = k$. ∎

Dynamic Programming

a. Consider the unweighted directed graph shown below. Does this graph has optimal substructure to find the longest simple path, say between q and r or q and t? Explain your answer. Can you use Dynamic Programming for this problem. Explain why or why not.

Ans.

No, this graph does not have optimal substructure. Consider

path $q \rightarrow r \rightarrow t$, which is a longest simple path from q to t. Is $q \rightarrow r$ a longest simple path from q to r? No, for the path $q \rightarrow s \rightarrow t \rightarrow r$ is a simple path that is longer. Is $r \rightarrow t$ a longest simple path from r to t? No again, for the path $r \rightarrow q \rightarrow s \rightarrow t$ is a simple path that is longer.

Since the graph does not have optimal substructure, we cannot use Dynamic Programming for this problem.