1) A 5-pointed-star in an undirected graph is a 5-clique. Show that 5-POINTED-STAR ∈ P, where 5-POINTED-STAR = { <G> | G contains a 5-pointed-star as a subgraph }.

Solution
We will show that 5-POINTED-STAR is in P by creating an algorithm that solves the problem in polynomial time in the size of the input. (For simplicity, we’ll judge the run time in terms of n, the number of vertices in the input graph G. If our run-time is polynomial in this, it’s certainly polynomial in the size of the input, which is larger.)

Given a graph G with n vertices, simply iterate through all \( \binom{n}{5} \) combinations of vertices in G. For each of these combinations, simply check if all 10 possible edges between the five vertices are present. If this is the case, answer YES. If this isn’t true for ANY of the \( \binom{n}{5} \) combinations of vertices, answer NO.

This algorithm correctly solves the problem using brute force and runs in \( O(n^5) \) time since \( \binom{n}{5} = \frac{n(n-1)(n-2)(n-3)(n-4)}{120} \) and looking up the existence of 10 edges should take \( O(1) \) time. Thus, TRIANGLE is in P.

2) Let CONNECTED = { <G> | G is a connected unweighted graph }. Show that CONNECTED ∈ P.

Solution
Choose an arbitrary vertex, v, in the graph. From v, run a Depth First Search or Breadth First Search, marking each vertex that is reachable from v. Then, check if each vertex is marked. If so, answer YES. Otherwise, answer NO. Both of the searches described run in \( O(V+E) \) time, which is polynomial in the size of the input.
3) Let $2\text{-SAT} = \{ <\phi> | \phi \text{ is a satisfiable 2-CNF formula} \}$. ($2\text{-CNF}$ is the same as $3\text{-CNF}$, but with only 2 variables allowed per clause.) Show that $2\text{-SAT} \in P$.

**Solution**
The status of each variable in the formula will be one of three things: UNASSIGNED, ASSIGNED, or LOCKED. The latter two assignments will be coupled with either TRUE or FALSE.

Initialize all variables to UNASSIGNED.

For each variable $(v_i)$, do the following:

1. Mark $v_i$ as ASSIGNED and assign it TRUE.
2. Mark off all the clauses that are true because of this assignment.
3. Look at all the clauses that have $v_i$ as FALSE and assign the other variable listed to be ASSIGNED and set to TRUE, so long as they were previously UNASSIGNED.
4. If any of these variables were already ASSIGNED, make sure no conflict has formed.
   If there is a conflict, LOCK $v_i$ to be FALSE and continue to the next loop iteration.
5. If no conflicts are found, continue the process until no new variables are forced to be ASSIGNED. If we end up with each variable either LOCKED or ASSIGNED, answer YES.

4) Let $\text{DOUBLE-SAT} = \{ <\phi> | \phi \text{ has at least two satisfying assignments} \}$. Show that $\text{DOUBLE-SAT}$ is NP-Complete by giving a reduction from $3\text{-SAT}$ to $\text{DOUBLE-SAT}$.

**Solution**
First, we note that $\text{DOUBLE-SAT}$ is in NP. One could provide a certificate of two satisfying assignments for the boolean formula and both could be verified in polynomial time in the size of the input.

Next, we’ll reduce from $3\text{-SAT}$ to $\text{DOUBLE-SAT}$. Create a new boolean formula $P'$ where $P' = \phi \land (x \lor \neg x \lor x)$, where $x$ is a variable not present in $\phi$. If $\phi$ has a satisfying instance, then $P'$ has at least two satisfying instances, one with the original variable settings and $x = \text{true}$ and the other with the original variable settings and $x = \text{false}$. Similarly, if $\phi$ has no satisfying assignment, $P'$ will NOT have any satisfying assignments, since $x$ isn’t the bottleneck in the formula. (Namely, adding clauses to a formula that is NOT satisfiable will never make it satisfiable.)
5) Let HALF-CLIQUE = \{ <G> | G is an undirected graph having a complete subgraph with at least n/2 nodes, where n is the number of nodes in G \}.

Show that HALF-CLIQUE is NP-complete.

**Solution**

We will prove that HALF-CLIQUE is NP-Complete by showing that (a) it is in NP, (b) CLIQUE reduces to HALF-CLIQUE in polynomial time.

HALF-CLIQUE is in NP because we can verify a certificate (n/2 or more vertices that form a clique) in polynomial size of the input graph. (We just have to go through each possible edge amongst the vertices to verify that they are all there. At most there are $O(n^2)$ pairs of vertices to check.)

Now, we reduce CLIQUE\(<G,k>\) to HALF-CLIQUE\(<G>\).

Our reduction will be split into two parts based on whether or not \(k \geq n/2\) or not.

(a) If \(k \geq n/2\), carry out the reduction as follows:

Create a new graph \(G'\) that is identical to \(G\), but simply add \(2k - n\) vertices to it that all have degree 0. (This means that these new vertices aren’t connected to any other vertices in the graph.) Thus, \(G'\) has \(n + (2k - n) = 2k\) vertices in it. This graph will satisfy HALF-CLIQUE if and only if it has a clique of size \(k\) or greater.

Clearly, if \(G\) has a clique of size \(k\) or greater, \(G'\) will also have a clique of size \(k\) or greater, where \(k\) is exactly half of the nodes of \(G'\). Similarly, if \(G\) has NO clique of size \(k\), \(G'\) can’t either because just adding vertices to \(G\) without any connections can not create new cliques that are bigger than the original cliques that \(G\) had.)

Thus, for cases where \(k \geq n/2\), we have reduced CLIQUE to HALF-CLIQUE because our graph \(G'\) has a HALF-CLIQUE if and only if \(G\) has a CLIQUE of size \(k\).

(b) Now, we consider the case where \(k < n/2\).

Create a new graph \(G'\) by taking \(G\) and adding \(n - 2k\) vertices to it. Call the original set of \(n\) vertices \(S\) and this new set of \(n - 2k\) vertices \(T\). Thus, \(G'\) has \(n + n - 2k = 2n - 2k = 2(n-k)\) vertices exactly.

Connect all pairs of vertices in \(T\) so that these form a complete subgraph. Next, connect all pairs of vertices with one vertex in \(S\) and one in \(T\). This completes the construction of \(G'\).

If \(G\) has a clique of size \(k\), we can see that \(G'\) will have a clique of size \(k + (n - 2k) = n - k\), because all \(k\) vertices from \(G\) that form a clique are connected to the \(n - 2k\) vertices in \(T\) that are all connected as well. If \(G'\) has a clique of size \(n - k\) (or larger), then it follows that \(G'\) satisfies HALF-CLIQUE.
If G has NO clique of size k, then it’s impossible for G’ to have a clique of size n – k. To see this, consider the largest CLIQUE in G’. It must have some vertices from set S and will necessarily have all vertices from set T, since these vertices are connected to all others in the graph. But, since no k vertices in S are all connected, the most number of vertices that could come from S is k – 1. Adding these vertices to the n – 2k vertices in T, we get n – k – 1, which is less than half of the vertices in G’. Thus, if G has no clique of size k, G’ does NOT satisfy HALF-CLIQUE, as desired.

This concludes the proof. In both cases, k ≥ n/2 and k < n/2, we have shown how to construct a graph G’ such that if and only if G has a CLIQUE of size k does G’ have a HALF-CLIQUE.
6) Let SUBSET-SUM-k = \{ <S, t, k> | S is a set of positive integers, such that there exists a subset B of S of size k, such that the sum of the elements in B is equal to t, the target. \}

Prove that SUBSET-SUM-k is NP-Complete by reducing SUBSET-SUM to it.

**Solution**

SUBSET-SUM-k is in NP since we can verify that a subset of size k adds up to the target t. Now, we will show that SUBSET-SUM-k is also NP-Complete.

Consider an instance of SUBSET-SUM with the set $S = \{x_1, x_2, \ldots x_n\}$ and a target $T$. Let $X = x_1 + x_2 + \ldots + x_n$. Here is an instance of SUBSET-SUM-k that is accepted if and only if the original instance of SUBSET SUM is:

$S' = \{X, X, \ldots X, X - x_1, X - x_2, \ldots X - x_n\}$, $T' = nX - T$, $k = n$

In particular, our set $S'$ has $2n$ elements in it: $n$ X’s and then one element corresponding to each element in the original set S. First of all, IF there is a subset in S that adds up to T, we can find a subset of $S'$ of size n that adds up to $T'$. (To do this, simply pick each term of the form $X - x_i$, that corresponds to the elements in the subset of S that add up to T. Then pick the remainder of the elements to be S’s. Based on this construction, the sum will be:

$$X + X + \ldots + (X - x_i), \text{ for each } x_i \text{ in the original subset adding up to } T.$$  

$$= nX - (x_i, \text{ for each } x_i \text{ in the original subset adding up to } T.)$$  

$$= nX - T$$  

$$= T', \text{ as desired.}$$

Now, we must show the other direction. If there is a subset in $S'$ of size $n$ that adds up to $T'$, then there will be a subset in S that adds up to T. Consider any subset of $T'$ of size $n$ that adds up to $nX - T$. Naturally, some of these values, say m of them, will be X, while the other $n - m$ will be of the form $X - x_i$. Adding these up we get that

$$mX + (n - m)X - (\text{the sum of } n - m \ x_i) = nX - T.$$  

Simplifying, we find that

$$(\text{the sum of } n - m \ x_i) = T, \text{ but this implies that there is a subset of } S \text{ such the sum of its elements equals } T, \text{ as desired.}$$

Thus, we have shown how to reduce SUBSET-SUM to SUBSET-SUM-k and can conclude that SUBSET-SUM-k is NP-Complete.