We must show two things, namely that Traveling Salesman (TSP) is in NP and that TSP is NP-Hard. These requirements follow from the definition of NP-Completeness.  
  
Note that NP is the complexity class of decision problems, thus we need the decision version of TSP. That is, given a graph G=(V,E) and a number k, does there exist a Tour C on G such that the sum of the edge weights for edges in C is less than or equal to k.  
  
Part 1 : TSP is in NP.  
Proof : Let a hint S be a sequence of vertices V = v1,...,vn. We then check two things, namely that every edge traversed by adjacent vertices is an edge in G (usually not required, since G is traditionally a complete graph), such that the sum of these edge weights is less than or equal to k. Secondly we check that every vertex in G is in V, which assures that every node has been traversed. We accept S if and only if S satisfies these two questions, otherwise reject. Both of these checks are clearly polynomial, thus our algorithm forms a verifier with hint S, and TSP is consequently in NP.  
  
Part 2: TSP is NP-Hard.  
Proof: To show that TSP is NP-Hard, we must show that every problem y in NP reduces to TSP in polynomial time. To do this, consider the decision version of Hamiltonian Cycle (HC). It is well known that HC is NP-Complete, thus HC is NP-Hard and every problem y in NP reduces to HC in polynomial time. If we reduce HC to TSP in polynomial time, then we will have shown that every y in NP reduces to TSP is polynomial time, since the sum of two polynomials is also a polynomial.  
Recall that the decision version of HC is : Given a graph G=(V,E), does there exist a simply cycle in G that traverses every vertex exactly once? Now proceed to observe that a simple cycle on n vertices has n edges. Now to reduce HC to TSP, employ the following algorithm.  
Take G=(V,E), set all edge weights equal to 1, and let k = |V|=n, that is, k equals the number of nodes in G. Any edge not originally in G then receives a weight of 2 (traditionally TSP is on a complete graph, so we need to add in these extra edges). Then pass this modified graph into TSP, asking if there exists a Tour on G with cost at most k. If the answer to TSP is YES, then HC is YES. Likewise if TSP is NO, then HC is NO.  
  
Proof: We must show that the reduction takes polynomial time and that solutions for HC are in 1-1 correspondence with solutions to TSP using the reduction. Clearly, the reduction takes polynomial time, so we are left with the latter.  
  
First Direction : HC has a YES answer => TSP has a YES answer.  
Proof: If HC has a YES answer, then there exists a simple cycle C that visits every node exactly once, thus C has n edges. Since every edge has weight 1 in the corresponding TSP instance for the edges that are in the HC graph, there is a Tour of weight n. Since k=n, and given that there is a Tour of weight n, it follows that TSP has a YES answer.  
  
Second Direction : HC has a NO answer => TSP has a NO answer.  
Proof: If HC has a NO answer, then there does not exist a simple cycle C in G that visits every vertex exactly once. Now suppose TSP has a YES answer. Then there is a tour that visits every vertex once with weight at most k. Since the Tour requires every node be traversed, there are n edges, and since k=n, every edge traversed must have weight 1, implying that these edges are in the HC graph. Then take this tour and traverse the same edges in the HC instance. This forms a Hamiltonian Cycle, a contradiction.  
  
This concludes Part 2. Since we have shown that TSP is both in NP and NP-Hard, we have that TSP is NP-Complete, as required.