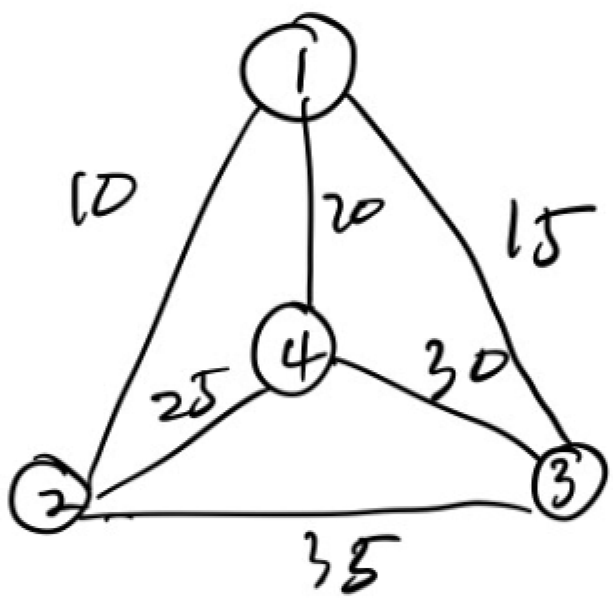
From the question, we can know that the problem is finding a Hamiltonian Cycle with lowest weight. Then we can have the following cycle as showed:



I will try to use DP to solve this problem.

We first let the number of vertices given taht the set be 1,2,3,4 and so on. Then we'll recall that 1 is the start and end of the output. We will try to use 1 as the start point to find the lowest cost path for each vertex i and it as the finished point factor, all vertices executing just once.
Then, we say the value of such a route is cost(i) and the price of the matching Cycle is value(i)+dist(i,1), where dist(i,1) is the distance between i and 1.
So, we just need to return the number which is the smallest of all [cost(i) + dist(i,1)].

Next, we need to connect recursively regarding of the sub troubles to determine the value(i) that utilizing DP. We can let C(s) which starts at 1 and finished at i and the i will be the value of the lowest cost path that passes over each vertex in the set s only once.
So, we just need to begin with all size 2-subsets and compute C(s), which i is for all subgroups and which s is the subgroup, so, we can have C(s) for each i of all size 3-subsets and so on.[Recursively]

Finally, we can clearly explain why APP $\leq$ 2*OPT.