ADA Mini HW #10

Student Name: 林楷恩
Student ID: b07902075

The procedure that "Find a node $v_j \in V \setminus S$ that is closest to S is just the same as how *Prim's Algorithm* select the next node to add to the MST. Let P_n be the Tree *Prim's Algorithm* obtains after the n-th iteration, and T_n be the tour after the n-th iteration of **NEAREST-ADDITION** algorithm. I show that $cost(T_n) \leq 2 \times cost(P_n)$ always holds by induction on n:

- **Base Case**: $n = 0 \implies$ Both T_0 and P_0 have only one edge (u, v)

$$\implies cost(T_0) = cost(P_0) \leq 2 \times cost(P_0)$$

- **Inductive Step**: Assume the statement holds when $n = m$, I show it also holds when $n = m + 1$.
  * v_i, v_j, v_k refer to the notation in given pseudo-code.

1. NEAREST-ADDITION: $cost(T_{m+1}) = cost(T_m) + w(v_i, v_j) + w(v_j, v_k) - w(v_i, v_k)$

2. Prim: $cost(P_{m+1}) = cost(P_m) + w(v_i, v_j)$

3. By *Triangle Inequality*, $w(v_i, v_j) + w(v_i, v_k) \geq w(v_j, v_k) \implies w(v_j, v_k) - w(v_i, v_k) \leq w(v_i, v_j)$
   $\implies w(v_i, v_j) + w(v_j, v_k) - w(v_i, v_k) \leq 2 \times w(v_i, v_j)$

4. By induction hypothesis, $cost(T_m) \leq 2 \times cost(P_m)$

5. By 3. and 4., $cost(T_{m+1}) \leq 2 \times cost(P_{m+1})$

Since $P_N = MST$, we have $cost(T_N) \leq 2 \times cost(MST)$. And in class, we have proved that the optimal tour H^* is formed by some tree T and some edge e, which indicates $cost(MST) \leq cost(H^*)$. Hence,

$$cost(T_N) \leq 2 \times cost(MST) \leq 2 \times cost(H^*)$$
$$\implies \text{the approximation ratio } \rho(N) = 2 \ \#$$