Nearest addition algorithm for metric TSP is a 2-approximation algorithm:

Proof: Nearest addition starts with min. cost edge (i_2, j_2), $S_2 = \{i_2, j_2\}$, tour: $i_2 - j_2 - i_2$.

~~Then finds node in N/S_2 s.t the distance~~

~~Then finds nodes j_3 in N/S_2 s.t. $c(i_3, j_3)$ is minimized. $i_3 \in S_2$~~

~~Suppose $i_2 = i_3$, then~~

Then finds edge (i_3, j_3) with $i_3 \in S_2$ and $j_3 \in N/S_2$ with $c(i_3, j_3)$ minimized.

Suppose $i_3 = i_2$, then in this iteration:

- we add edge (i_3, j_3)
- $S_3 = \{i_2 = i_3, j_3, j_2\}$
- tour: $i_2 - j_3 - j_2 - i_2$

And continues...

Prim's algorithm starts from an arbitrary node say i_2 and let's $S_1 = \{i_2\}$. Finds the min. cost edge (i_2, j_2) s.t $i_2 \in S_1$ and $j_2 \in N/S_1$.

Then $S_2 = \{i_2, j_2\}$, the finds the min cost edge (i_3, j_3) s.t. $i_3 \in S_2$ and $j_3 \in N/S_2$. Thus, $S_3 = \{i_2 = i_3, j_3, j_2\}$ and so on...

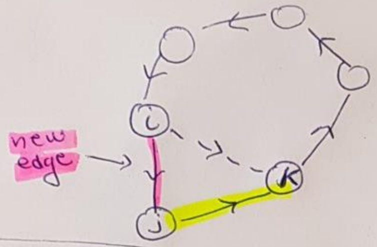
$$\boxed{\text{Sequence of edges: } \underbrace{(i_2, j_2)}_{\substack{\text{- tour size 2} \\ \text{- tree size 1}}}, \dots, \underbrace{(i_n, j_n)}_{\substack{\text{- tour size } n \\ \text{- tree size } n-1}} \text{ is the same}}$$

$F = \{(i_2, j_2), \dots, (i_n, j_n)\}$. is a MST

$$\Rightarrow OPT \geq \sum_{k=2}^{n} c(i_k, j_k).$$

- Cost of tour on first two nodes i_2, j_2 is $2c(i_2,j_2)$

- In future iterations, suppose a city j is inserted between cities i and k:



How much does the length of the tour increase?

$$\underbrace{c_{ij} + c_{jk} - c_{ik}}_{T.I.} \leq$$

$$\leq c_{ij} + c_{ij} = 2c_{ij}$$

$$\boxed{\begin{aligned} \text{By T.I.: } & c_{jk} \leq c_{ji} + c_{ik} \\ \iff & c_{jk} - c_{ik} \leq c_{ji} \end{aligned}}$$

Thus, it increases at each iteration ℓ by at most $2c(i_\ell, j_\ell)$.

$\therefore$ Cost of tour $\leq 2 \sum_{\ell=2}^{n} c(i_\ell, j_\ell) \leq 2OPT$. // $\square$