1. In the nearest neighbour approximation to the traveling salesman problem, we begin by selecting an arbitrary city as starting point

- Let, the edge lengths of a tour selected with this algorithm labeled $l_1 \geq l_2 \dots \geq l_n$ (n = edges)

where $\sum_{i=1}^{n} l_i = HEUR.$

- The optimal solution by, triangle inequality is,

$$OPT \geq 2 l_1 \quad \text{———} \; ①$$

l_1 = endpoint should be visited during optimal tour

- Take $i = k+1$ where k = number of cities we know (From Professor Notes)

$$OPT \geq 2 \sum_{i=k+1}^{2k} l_i \quad \text{———} \; ②$$

- So now when n is odd and $n = 2k+1$, where $k \in$ Natural numbers set ($N+$).

- Therefore,

$$T_n = T_{2k+1} \geq \sum \{l_i, l_j\}$$

- Replacing first k edges l_i with members of last k edges l_i, we get from eqn ① & ②

$$T_n \geq 2 \sum_{i=k+2}^{2k+1} l_i + l_{k+1}$$

$$T_n > 2 \sum_{i=k+2}^{2k+1} l_i$$

Considering $k = \frac{n-1}{2}$ we get

$$T_n > 2 \sum_{i=\frac{n-1}{2}+2}^{2\times\frac{n-1}{2}+1}$$

$$T_n > 2 \sum_{i=\lceil \frac{n}{2} \rceil +1}^{n} l_i$$

(2) We know,

$$\sum_{i=1}^{n} l_i = HEUR \quad \text{for edge lengths } \{l_1, l_2 \ldots l_n\}$$

- The optimal solution by triangle inequality is

$$OPT \geq 2\, l_i$$

- The above relation applys to the Nearest - Neighbour heuristics for the Traveling salesman Problem.
- Their will be equal increase in the OPT if we increase the length of the edge.
- In Professor's notes we have seen that

$$\frac{HEUR}{OPT} \leq \frac{[\log n] + 1}{2}$$

- So there would be increase in HEUR iff OPT has an increase.
- Also formula states that the ratio $\frac{HEUR}{OPT}$ is directly propotional to log n where n is the number of [redacted] edges
- Thus it is clear that ratio is totally

DATE :

2. dependend on the number of [redacted] edges and not on the length of edges.

- Hence, we can say that an increase in the length of last edge will not bring any major change in the ratio HEUR / OPT.

- Hence Proved.

3. (a) For any 3 nodes i, j, k the triangle inequality holds for the edges between them.

$$c_{ij} \leq c_{ik} + c_{kj} \quad \forall\, i, j, k \in V$$

- Suppose a triangle not satisfying triangle inequality property
- Let's add sufficiently large number to each element of cost matrix. Consider value to be added be 'X' to the weight of each element / edge in the graph.
- If the graph composes of n nodes, the total solution weight will increase by $X * n$ for all solutions.
- Adding the constant weight to every element won't chang the path TSP solution takes as equal weight is added all over.
- Assume — Edges $(\{i,j\}, \{i,k\}, \{k,j\})$ in a graph that violets the triangle inequality, then

$$q_{ikj} = c_{ij} - c_{ik} - c_{kj} \quad \text{—— (1)}$$

We choose the largest value of q_{ikj} over all edge triples as the value X that we will use.
- $c'_{ij} \rightarrow c_{ij} + X$ for edge (i, j) (arbitary edge)

$$c'_{ij} = c_{ij} + X \;;\quad c'_{ik} = c_{ik} + X \;;\quad c'_{kj} = c_{kj} + X$$

- After substitution of X value in eq^n (1) we get;

$$c'_{ik} + c'_{kj} = c_{ik} + X + c_{kj} + X$$
$$= c_{ik} + c_{kj} + 2X \geq c_{ij} + q$$
$$= c_{ik} + c_{kj} + 2X \geq c'_{ij}$$

3. (a) This shows adding X weight equally to all edges removes triangle inequality violations in any graph.

- We can reduce TSP without triangle inequality to a TSP with triangle inequality in polynomial time.
- Hence proved that given an arbitrary symmetric cost matrix C of inter-city distances, it is possible to force the triangle inequality to hold by adding sufficiently large value equally to all elements in the graph.

* **Reference** - cs. standford - edu /people / levisan

3. (b) In class we have seen that if the arbitary symmetric cost matrix holds the triangle inequality then and then only the bound on quality of closet insertion heuristics holds

- So now if the cost matrix doesn't hold the triangle inequality then it can't gaurantee the bound on quality of closest insertion heuristic will hold.
- Finding Approximation of TSP is NP-hard if the symmetric matrix do not hold the triangle inequality
- Thus we cannot guarantee the approximation of TSP can be found.

4. (a) b.

$$\frac{\text{cost of cheapest insertion}}{\text{cost of optimal tour}} < 2$$

- Let cost of cheapest insertion be denoted as INSERTION & cost of optimal tour be OPT

$$\therefore \frac{\text{INSERTION}}{\text{OPT}} < 2 \quad \text{—— (1)}$$

- Consider following lemma to prove eq^n (1)
- Suppose for a traveling salesman graph (N, d) with n nodes, a tour length INSERTION could be constructed by insertion method.
- Then tour T_i and node a_i selected by this insertion method will satisfy

$$\text{COST}(T_i, a_i) \leq 2d(p,q) \quad \text{—— (2)}$$

where,

~~all nodes~~ p for all nodes p and q such that p is in T_i and q is not in T_i and $1 < i < n$

- Consider TREE is the length of minimal spanning tree for (N, d) then,

$$\text{INSERTION} \leq 2 \cdot \text{TREE} \quad \text{—— (3)}$$

PROOF :

- M be the minimal spanning tree that connects all vertices without any cycles
- In order to insert a_i into tour T_i the corresponding edge of M will have one endpoint in Tour T_i and other in $N - T_i$
- From eq^n (2) we can show the cost of each step is no more than 2 times the corresponding edge
- For each node in MST a_i with $i > 0$, implies node a_i is compatible with node a_j if $j < i$ and there is unique path in M connecting

4. (a) each pair of nodes. For each node a_i with $i>0$, all the immediate nodes in the unique path in M connecting a_i & a_j have indices greater than i.

- Thus a_j is the first node in T_i in path a_i to a_j
- Critical Node - Node with largest index compatible with a_i where $i>0$
- Critical Path - For a_i is the unique path in M between a_i and its critical node.
- Critical Edge - an edge in critical path whose one endpoint is critical node and other endpoint in $(N-T_i)$

No two Nodes can have same critical edge.

- Assume on contrary that a_i and a_j $(j>i)$ have same critical edge.
- With $x>z$, let two endpoints of this critical edge be a_x and a_z.
- Node with lower index is critical node & node with higher index is on critical path
- Therefore node a_z is critical node for both a_i and a_j.
- Thus before reaching a_z the critical path ~~P~~ a_i to a_j pass through a_x
- Every edge in P belongs to either critical path for a_i or a_j or both.
- Every intermediate node on P has index greater than i because of above observation.
- Since Path P from a_j reaches a node of lower index (a_i) some node a_y along path is compatible with a_j
- Now $y \geq i$ because a_m is one path P and $i > z$ because a_k is compatible node for a_i

4.

(a) - This indicates $m > z$ and so a_g is compatible node for a_j with higher index than a_z.

- This contradicts the assumption. Therefore no two nodes can have the same critical edge.
- Thus given a minimal spanning tree we can associate a unique edge in that tree with each node inserted by insertion method.
- Consider $edge(i)$ be the critical edge for node a_i. Since one endpoint of $edge(i)$ is in T_i and other endpoint is not, therefore from eqn ②

$$COST(T_i, a_i) \le 2 \cdot d(edge(i))$$

$$\Downarrow$$

$$\sum_{i=1}^{n-1} COST(T_i, a_i) \le 2 \cdot \sum_{i=1}^{n-1} d(edge(i))$$

$$\Downarrow$$

implies eqn ③

- Equation ② holds the cheapest insertion method. For cheapest insertion there is for each i a node y_i in T_i such that

$$d(y_i, a_i) \le d(p, q) \quad \text{—— ④}$$

for all p in T_i and q in $N - T_i$

$$COST(T_i, a_i) \le 2\, d(y_i, a_i) \quad \text{—— ⑤}$$

$\Downarrow$ implies

$$COST(T_i, a_i) \le 2\, d(p, q).$$

* Reference - An analysis of several heuristics for TSP by DANIEL J ROSENKRANTZ (Professor's Notes)

4. (b) Each time a new city is added say k we need to find nodes $i, j \in$ Tour and $k \notin$ Tour.

- Thus Cost $(C_{ik} + C_{kj} - C_{ij})$ is minimized
- Each city/node can maintain a storage say min-heap storing the cost of inserting it to every edge of the tour and update the heap each time new node is inserted.
- As n number of cities/nodes will be inserted newly to the tour, we need to update heap for every insertion.
- Updating heap takes time complexity of $O(\log n)$.
- Therefore, total running time is $O(n^2 \log n)$

4. (c) A tree can be constructed from optimal tour by deleting its longest edge and this longest edge could have length at least OPTIMAL/n where n is the number of nodes in the ~~number~~ problem.

$$\text{TREE} \leq \text{OPTIMAL} - \frac{\text{OPTIMAL}}{n}$$

Since minimal spanning tree is no longer than this tree.

$$\therefore \text{TREE} \leq \text{OPTIMAL} * \left(1 - \frac{1}{n}\right)$$

this implies;

$$\frac{\text{INSERTION (Cheapest)}}{\text{OPTIMAL}} \leq 2\left(1 - \frac{1}{n}\right)$$