Section 2.4: TSP

The Traveling Salesman Problem (TSP)

Input: Weighted complete graph G

$$c_{ij} = c_{ji}, \qquad i,j \in V$$
$$c_{ii} = 0, \qquad i \in V$$
$$c_{ij} \geq 0, \qquad i,j \in V$$

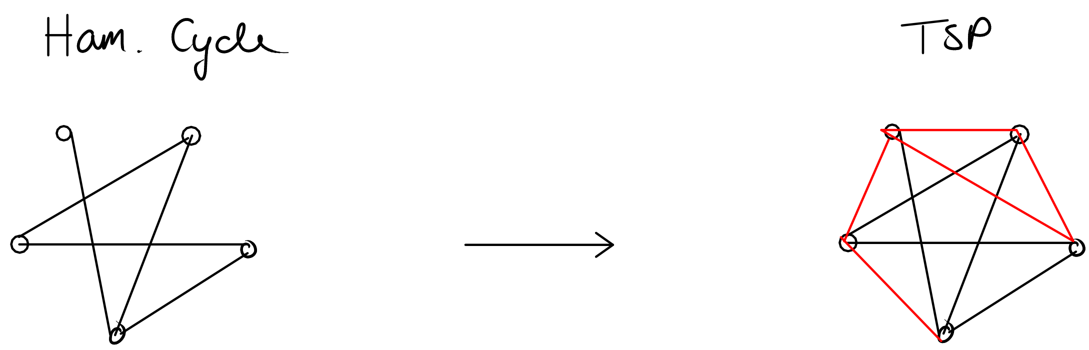
Output: $\underbrace{\text{Hamilton cycle}}_{\text{Cycle visiting each vertex exactly once.}}$ of min. total weight

No approximation guarantee possible:

Theorem 2.9

$\nexists$ α-approx alg. for TSP, for any $\alpha > 1$

Proof: Reduction from Hamilton Cycle:



$c_{ij} = 1$, $c_{ij} = \alpha n + 1$

$\exists$ ham. cycle $\iff$ $\exists$ tour of cost n

$\iff$ α-approx. alg. gives tour of cost $\le \alpha n$

$\iff$ $\nexists$ red edge in the tour

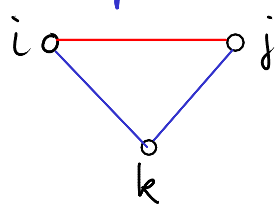
$\square$

Note: The proof does not require α to be a constant. In fact, it could be 2^n, or <u>any</u> function computable in poly. time.

Metric TSP:

The edge weights satisfy the triangle inequality:

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



For metric TSP, the proof of Thm 2.9 does not work (the max. possible cost of the red edges would be 2).
We will see a 2- and a 3/2-approx. alg. for Metric TSP.

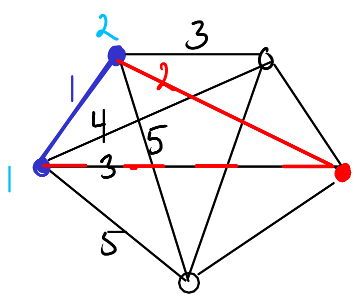
For the metric TSP problem, we will consider three algorithms:

The Nearest Addition algorithm	2-approx.
The Double Tree algorithm	2-approx.
Christofide's Algorithm	3/2-approx

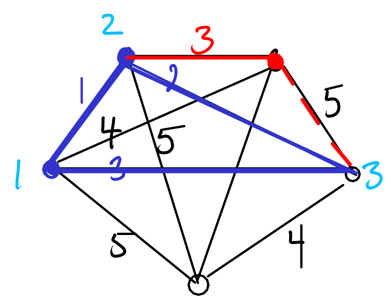
Nearest Addition (NA)

$u, v \leftarrow$ two nearest neighbors in V
Tour $\leftarrow \langle u, v, u \rangle$
For $i \leftarrow 1$ to $n-1$
    $v \leftarrow$ nearest neighbour of Tour
    $u_1 \leftarrow$ nearest neighbor of v in Tour
    $u_2 \leftarrow u_1$'s successor in Tour
    Add v to Tour between u_1 and u_2

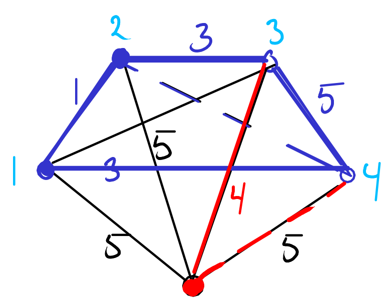
Ex:



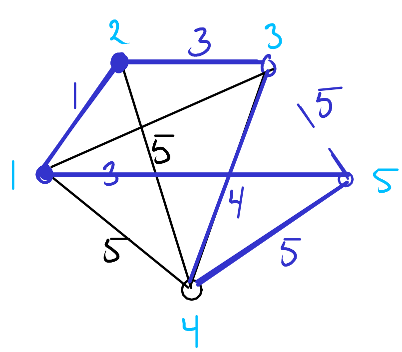
add new vertex $\longrightarrow$ between 2 and 1



add new vertex $\longrightarrow$ between 2 and 3



add new vertex $\longrightarrow$ between 3 and 4

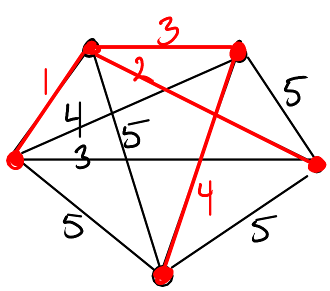


$C = 1+3+4+5+3$
$= 16$

Nearest Neighbor is a 2-approx. alg.:
We will prove that

$$\boxed{\begin{aligned} C_{NA} &\le 2 \cdot c(MST) \\ c(MST) &\le C_{OPT} \end{aligned}}$$ (Lemma 2.10)

The solid red edges are exactly those chosen by Prim's Algorithm:

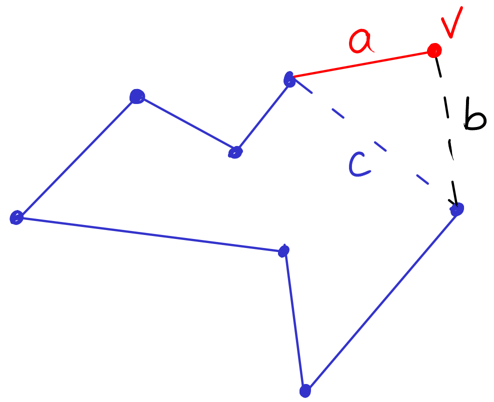


$C = 1+2+3+4 = 10$

Thus, the total cost C of these edges is that of a minimum spanning tree:

$$\boxed{C = c(MST)}$$

Adding a new vertex v to the tour, we add two edges and delete one:



Adding v costs

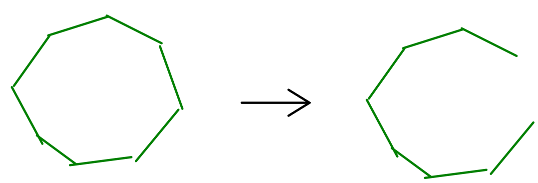
$a+b-c \leq a+(a+c)-c = 2a$

by the triangle inequality

Thus,

$$\boxed{C_{NN} \leq 2C = 2c(MST)}$$

Deleting any edge from a tour, we get a spanning tree T:



Thus, $\boxed{C_{OPT} \geq c(T) \geq c(MST)}$

Now,

$$\left\{\begin{array}{l} C_{NA} \leq 2 \cdot c(MST) \\ c(MST) \leq C_{OPT} \end{array}\right.$$

$\Downarrow$

$$C_{NA} \leq 2 \cdot C_{OPT}$$

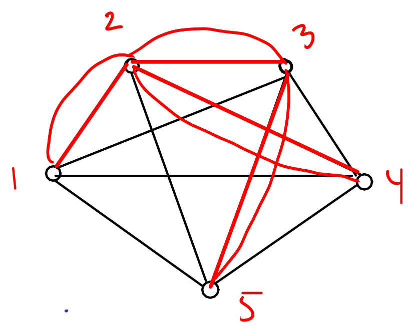
This proves:

Theorem 2.11

Nearest Addition is a 2-approx. alg.

Double Tree algorithm

Noting that NA adds the edges of a MST one by one, we could also make a MST T and traverse T, making short cuts whenever we would otherwise visit a node for the second time:

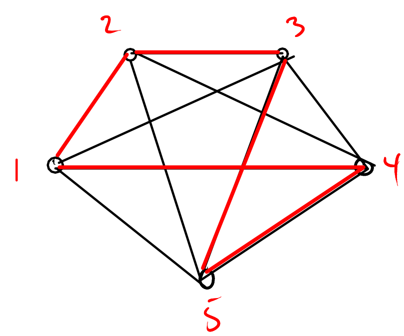


By the triangle inequality, this distance is no longer

$\langle 1, 2, 3, 5, 3, 2, 4, 2, 1 \rangle$

than this total distance

$\downarrow$



$\langle 1, 2, 3, 5, 4, 1 \rangle$

An Euler tour is a traversal of a graph that traverses each edge exactly once.

A graph that has an Euler tour is called eulerian.

A graph is eulerian if and only if all vertices have even degree.

Constructive proof of „if" in exercises for Monday.

Double Tree Algorithm (DT)

$T \leftarrow MST$

Double all edges in T

Make Euler tour ETour

Tour $\leftarrow$ vertices in order of first appearance in ETour

Same analysis as for NA:

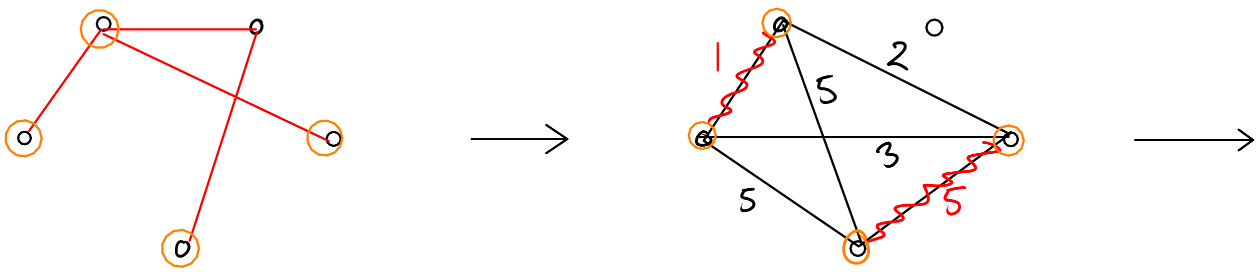
$$c_{DT} \leq 2\, c(MST) \leq 2 \cdot c_{OPT}$$

Theorem 2.12

Double Tree is a 2-approx. alg

Christofide's Algorithm

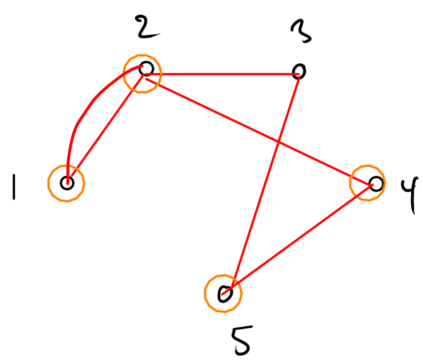
Next idea:
Not necessary to add n-1 edges to obtain even degree for all vertices. Instead: add a min. perfect matching on vertices of odd degree in the MST



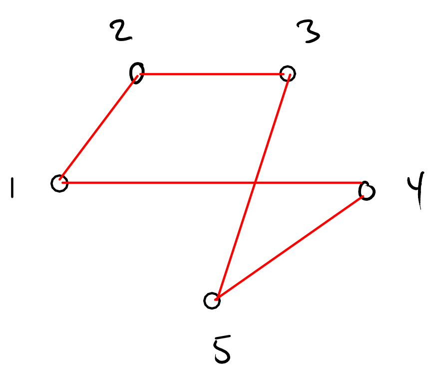
MST

Odd degree

Min. matching



Euler tour : $\langle \underline{1}, \underline{2}, \underline{3}, \underline{5}, \underline{4}, 2, \underline{1} \rangle$



TSP tour : $\langle 1, 2, 3, 5, 4, 1 \rangle$

Christofide's Algorithm (CA)

$T \leftarrow$ MST

$M \leftarrow$ minimum perfect matching on odd degree vertices in T

ETour $\leftarrow$ Euler tour in the subgraph $(V, E(T) \cup M)$

Tour $\leftarrow$ vertices in order of first appearance in ETour

Theorem 2.13

Christofide's Algorithm is a $3/2$-approx. alg.

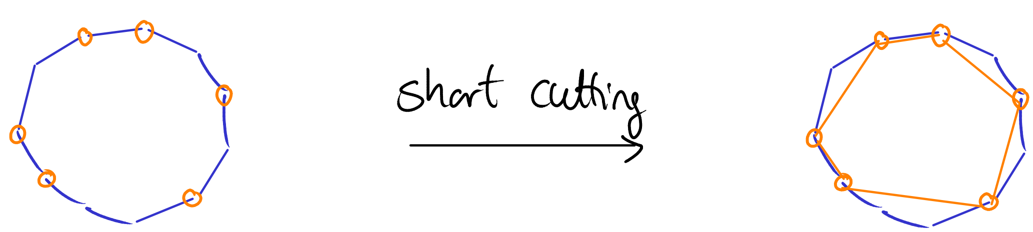
Proof:

By the triangle inequality,

$$c_{CA} \leq c(T) + c(M), \text{ where}$$

$$c(T) \leq c_{OPT}, \text{ by the arguments above}$$

Furthermore, $c(M) \leq \frac{1}{2} c_{OPT}$:



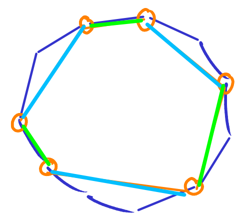
Optimal TSP tour

o : Odd degree vertices in MST T

C : cost of orange cycle

$C \leq c_{OPT}$, by $\triangle$-ineq.

The cycle on the odd degree vertices consists of two perfect matchings:



$$C = C + C$$
$$\Downarrow$$
$$\min\{C, C\} \leq \frac{1}{2} \cdot C \leq \frac{1}{2} \cdot c_{OPT}$$

Since M is a minimum matching on the odd degree vertices,

$$c_M \leq \min\{C, C\} \leq \frac{1}{2} \cdot c_{OPT}$$

$\square$