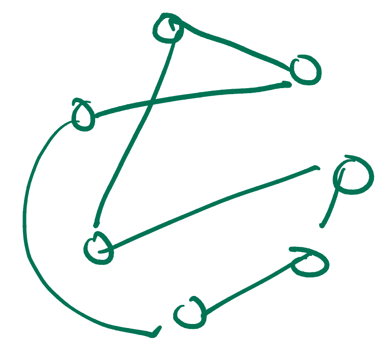
r-approximation $\quad r \geq 1$

$$\frac{cost(\tilde{S})}{OPT} \leq r \quad \text{or} \quad \frac{OPT}{cost(\tilde{S})} \leq r$$

min cost $\qquad$ max cost

$\hookrightarrow cost(\tilde{S}) \leq r \cdot \boxed{LB} \leq r \cdot OPT$ $\qquad \hookrightarrow cost(\tilde{S}) \geq \frac{\boxed{UB}}{r} \geq \frac{OPT}{r}$

2-approx for VC, MAX-CUT



TSP : Travel Salesperson Problem

n cities : numbered from 1 to n

D_{ij} = cost of going from city i to city j

tour permutation of the cities : $c_1\, c_2 \ldots c_n$

$$cost(tour) = \left(\sum_{i=1}^{n-1} D_{c_i, c_{i+1}}\right) + D_{c_n, c_1}$$

$\Rightarrow$ MIN-COST tour

TSP is NPC

▶ There is no r-approximation, $r > 1$, for TSP unless $P = NP$

↳ message: approximating TSP is as hard as solving it optimally

Suppose A_r is a poly-time algorithm that provides an r-approx for TSP

We build an algorithm A^H to solve $HAM \in NPC$

$A_H(G) \to$ yes if G is Hamiltonian
$\to$ NO o.w.

$G = (V, E)$

1) $D_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ \underline{1 + r|V|} & \text{otherwise} \end{cases}$

gap technique (boost weigth suitably)

2) Execute A_r on D, and let C be the (approximated) cost returned by A_r

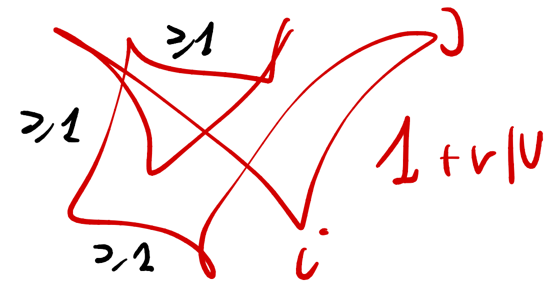
3) if $C \leq \underline{r \cdot |V|}$ then we answer YES
else we answer $\boxed{\text{NO}}$

A_r poly-time
$\Rightarrow A^H$ poly-time

It works for the following reason:

G is Hamiltonian: $\exists$ tour in D that has all costs to 1 as the Hamiltonian cycle uses edges in E $\Rightarrow$ $\text{cost}(\text{tour}) = |V|$ $\Rightarrow$ $C \leq r \cdot \text{cost}(\text{tour})$
$= r \cdot |V|$

G is not Hamiltonian: $\forall$ tour in D, there should be at least one distance $D_{ij} = 1 + r|V|$ as there is at least one $(i,j) \notin E$ that is "traversed" by the tour $\Rightarrow$ <u>any</u> tour has cost $\geq (|V|-1) \times 1 + \underline{(1 + r|V|)} > \underline{r|V|}$



METRIC TSP : $D_{ij} + D_{jk} \geq D_{ik}$ triangle inequality

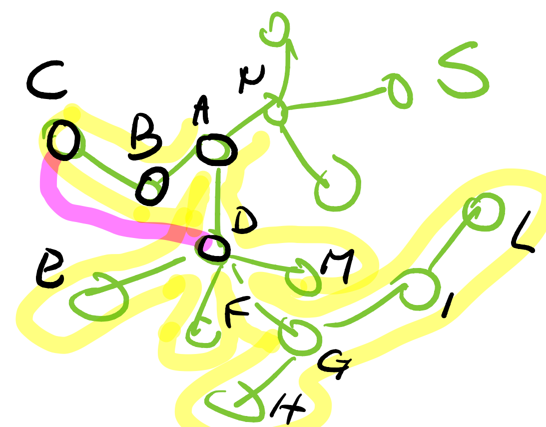
2-approx

MAIN IDEA let $G = (V, E, D)$ be the complete weighted graph whose edge weights are in D

1] Compute $S = MST(G)$
minimum-cost spanning tree

2] Compute a tour T as follows:
Perform a traversal of S :
each time we see a node in S
for the first time, we add it to T

$T = ABCDEF\ldots$
tour



1. $\text{cost}(OPT) \geq \overset{LB}{\text{cost}(S)}$

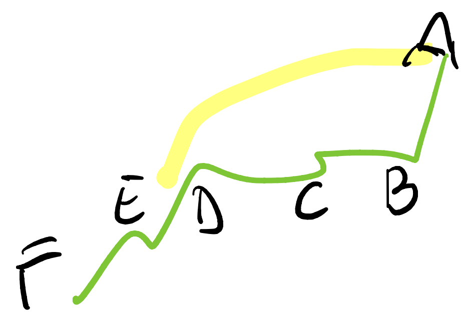
as a tour without an edge is a spanning tree

2. T is the approximated solution

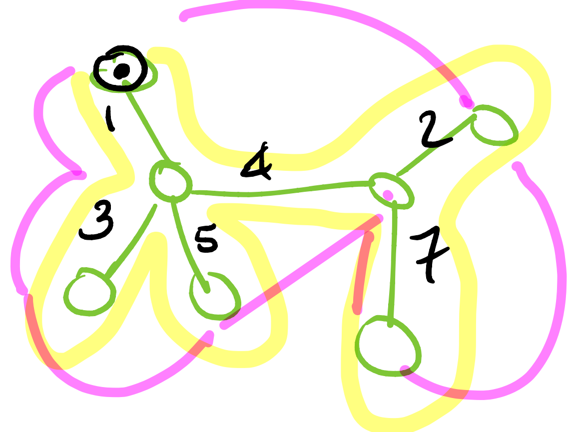
$\text{cost}(T) \leq 2 \cdot \text{cost}(S)$

by triangle inequality + each edge weigth is summed twice in $2 \cdot \text{cost}(S)$

$D_{A,E} \leq D_{AB} + D_{BC} + D_{CD} + D_{DE}$



① + ② $\Rightarrow \text{cost}(T) \leq 2 \cdot \text{cost}(S) \leq 2 \cdot OPT$



$\text{cost} = 1\ 3\ 3\ 5\ 5\ 4\ 7\ 7\ 2\ 2\ 4\ 1$ Euler

$\text{cost}\ 1\ 3\ \underbrace{?}_{\le 3+5}\ \underbrace{?}_{\le 5+4}\ 7\ \underbrace{?}_{\le 7+2}\ \underbrace{?}_{\le 2+4+1}$ T

using triangle inequality