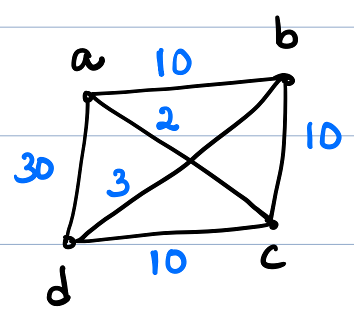
Travelling Salesman Problem

We are given a undirected complete graph G that has a non-negative integer cost c_{uv} associated with each edge $uv \in E(G)$, and we must find a hamiltonian cycle (a tour) of G with minimum cost.

For $A \subseteq E(G)$

$$c(A) = \sum_{uv \in A} c_{uv}$$

Example :



Hamiltonian cycle of minimum cost is

$a - b - d - c - a$ of cost $10 + 3 + 10 + 2 = 25$

Decision Version of TSP

I/P: Complete graph G, $c : V \times V \to \mathbb{Z}$, $k \in \mathbb{Z}$

Q: Does G has a Hamiltonian cycle with cost at most k.

Remark: TSP is NP-Complete

Proof Idea: Hamiltonian cycle $\leq_p$ TSP

Let $G=(V,E)$ be an instance of Hamiltonian cycle problem. We construct an instance of TSP as follows.

We form a complete graph $G' = (V, E')$, where $E' = \{ uv \mid u \in V, v \in V \}$. Cost function c is defined as

$$c_{uv} = \begin{cases} 0 & \text{if } uv \in E \\ 1 & \text{otherwise} \end{cases}$$

The instance of TSP is then $(G', c, 0)$

which can be constructed in polynomial time.

Claim: The graph G has a hamiltonian cycle if and only if graph G' has tour of cost at most zero.

Proof: Exercise

Therefore we now study a 2-approximation algorithm for TSP (special case).

Assumption :

We assume that the cost function c satisfies the triangle inequality.

ie, for all $u, v, w \in V(G)$

$$c_{uw} \le c_{uv} + c_{vw}$$

if costs on edges of G satisfies the triangle inequality then it is called Metric TSP.

Theorem: Metric TSP is NP-hard

Proof Hamiltonian cycle $\leq_p$ Metric TSP

Given a graph G, construct a complete graph G' on the vertex set $V(G)$, with cost function c defined as follows.

$$c_{uv} = \begin{cases} 1 & \text{if } uv \in E \\ 2 & \text{otherwise} \end{cases}$$

Clearly c satisfies triangle inequality.

Claim: The graph G has a hamiltonian cycle if and only if graph G' has tour of cost n.

Proof: Exercise

2-approximation algorithm for metric TSP

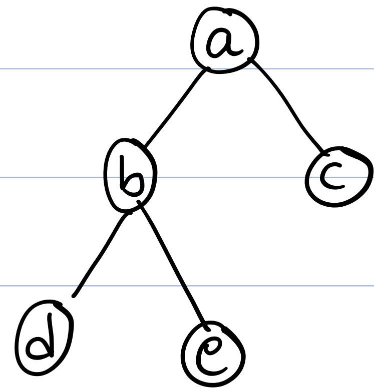
APPROX-TSP-TOUR(G, c)

select a vertex $r \in G.V$ to be a "root" vertex
compute a minimum spanning tree T for G from root r
  using MST-PRIM(G, c, r)
let H be a list of vertices, ordered according to when they are first visited
  in a preorder tree walk of T
return the hamiltonian cycle H

Recap: Tree Traversals

↓

walking the tree



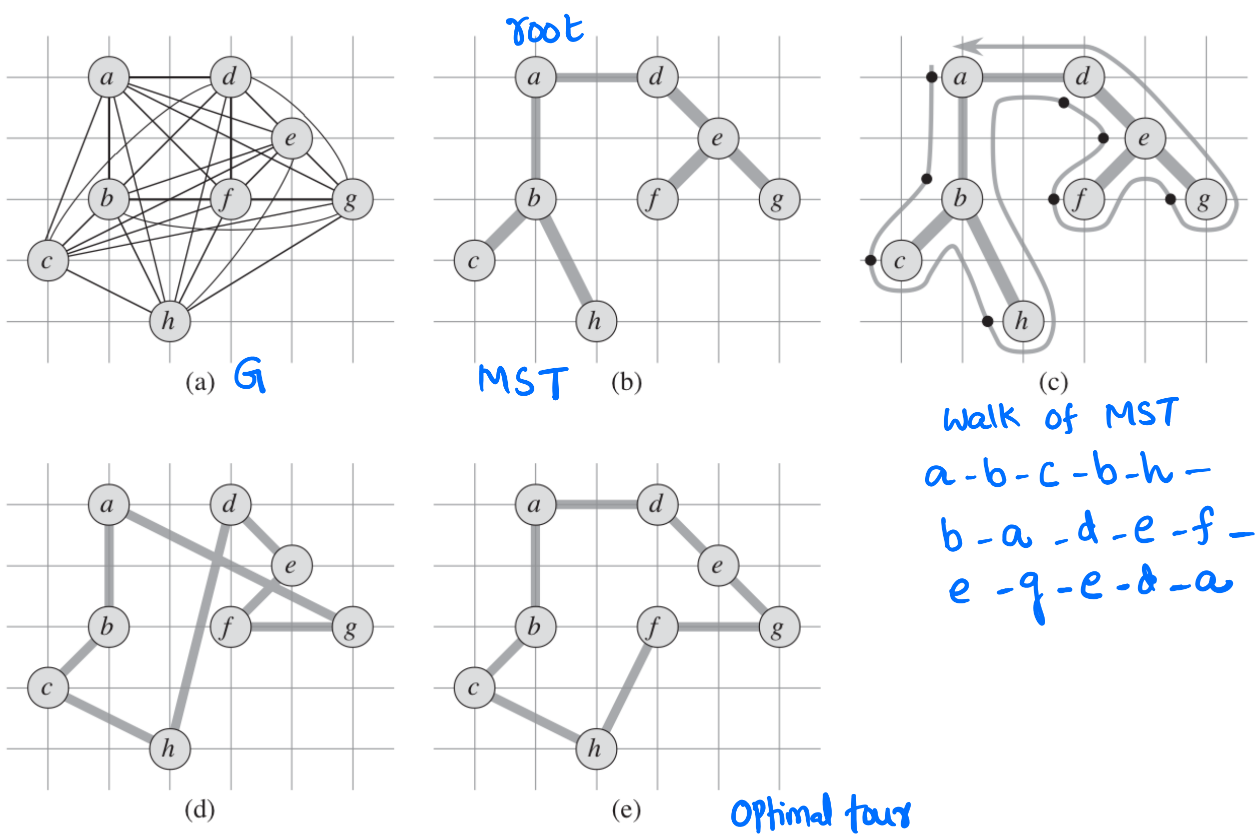
Inorder (left, root, right) : d b e a c

Preorder (root, left, right) : a b d e c

Postorder (left, right, root) : d e b c a

Illustration of APPROX-TSP-TOUR

cost function is Euclidean distance



(a) G MST (b) (c)

Walk of MST
a-b-c-b-h-
b-a-d-e-f-
e-g-e-d-a

(d) (e) Optimal tour

Tour obtained by visiting
the vertices in the order
given by Preorder walk
(O/P of the APPROX-TSP-TOUR)

Running time of APPROX-TSP-TOUR

APPROX-TSP-TOUR(G, c)

1 select a vertex $r \in G.V$ to be a "root" vertex $O(1)$

2 compute a minimum spanning tree T for G from root r using MST-PRIM(G, c, r) } Prim's algorithm $O(n^2)$

3 let H be a list of vertices, ordered according to when they are first visited in a preorder tree walk of T $O(n)$

4 **return** the hamiltonian cycle H $O(1)$

Analysis :

Let H^* denote an optimal tour for the given set of vertices.

We obtain a spanning tree by deleting any edge from a tour and each edge cost is non-negative.

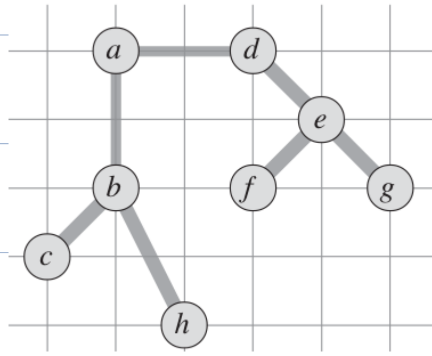
$\therefore$ the weight of the MST T computed in line 2 of the algorithm provides a lower bound on the cost of an optimal tour.

$$C(T) \leq C(H^*) \quad \text{——} \quad \mathrm{I}$$

Full walk W of T list the vertices when they are first visited and also whenever they are returned to after a visit to a subtree.

Full walk in the example is

a - b - c - b - h - b - a - d - e - f - e - g - e - d - a



ie, full walk traverses every edge of T exactly twice

$$\therefore \quad C(W) = 2\,C(T) \quad \text{— II}$$

From I & II

$$C(W) \le 2\,C(H^*) \quad \text{— III}$$

Unfortunately, the full walk W is generally not a tour, since it visits some vertices more than once.

Using the triangle inequality, we can delete a visit to any vertex from W and the cost does not increase.

Example

$$\underbrace{a - b - c - b - h}_{a - b - c - h} - \underbrace{b - a - d - e - f - e - g - e - d - a}$$

a - b - c - h

$c_{ch} \leq c_{cb} + c_{bh}$

By repeatedly applying this operation, we can remove from W all but the first visit to each vertex.

In our running example, we get

a - b - c - h - d - e - f - g

↳ Pre-order walk of the tree T.

Let H be the cycle corresponding to this preorder walk. It is a Hamiltonian cycle as every vertex is visited exactly once.

Since H is obtained by deleting vertices from the full walk W, we have

$$C(H) \leq C(W) \quad \text{—— IV}$$

from III & IV

we get $C(H) \leq 2\,C(H^*)$.

XX

Metric TSP $\frac{3}{2}$ approximation algorithm

First we recap few definitions.

may not be distinct

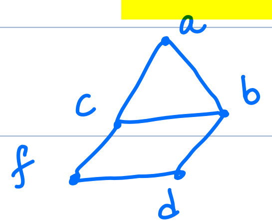
Def:- A walk in a graph is a sequence of vertices $v_1, v_2, \ldots v_k$ such that $v_i v_{i+1} \in E$ for $i = 1, 2, \ldots k-1$.

If the vertices in a walk are distinct then the walk is called a Path. If the edges in a walk are distinct then the walk is called a trial.

A cycle is a path $v_1, \ldots v_k$ together with the edge $v_k v_1$ $(k \geq 3)$.

A Trial that begins and ends at the same vertex is called a circuit (or) Closed trail

Ex



walk = a c f c b d

Trial = b a c b d

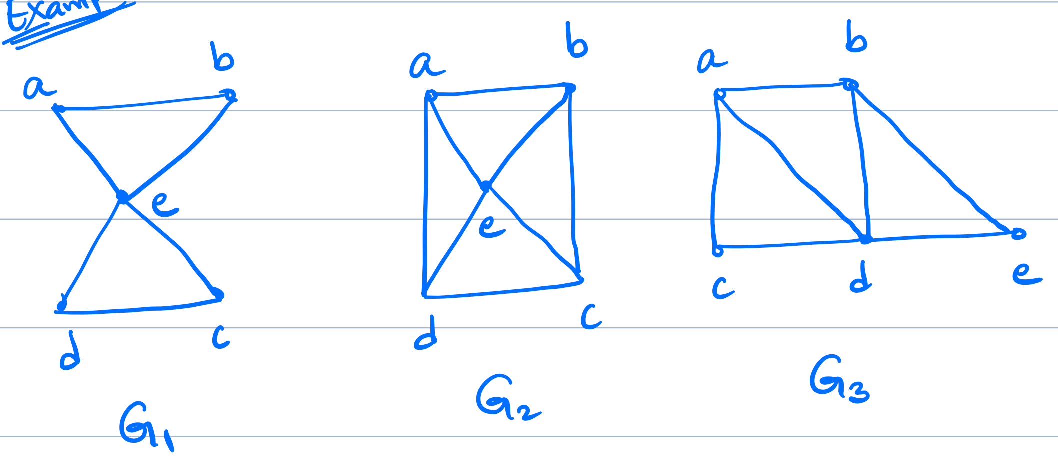
Path = a c f d

circuit = a c f d b a

Def :- A graph is Eulerian if it has a closed trail containing all edges. We call a closed trail a circuit when we do not specify the first vertex but keep the list in cyclic order.

An Eulerian circuit or Eulerian trail in a graph is a circuit or trail containing all the edges.

Example :



G_1 has an Euler circuit → a, e, c, d, e, b, a

G_2 & G_3 has no Euler circuit.

G_3 has an Euler trail → a, c, d, e, b, d, a, b

G_2 doesn't have Euler trail.

Definition:

A graph is called an Eulerian graph if it contains an Euler Circuit.

Theorem:- (Euler 1736) A connected graph G is Eulerian iff every vertex has even degree.

Pmf: Skipped. Refer to any text on graph theory

Back to Metric TSP

* We can improve the approximation factor to 3/2 by adding a minimum cost Perfect matching on vertices of odd degree in MST.

Algorithm 3.10 (Metric TSP – factor 3/2) [Christofides]

1. Find an MST of G, say T.
2. Compute a minimum cost perfect matching, M, on the set of odd-degree vertices of T. Add M to T and obtain an Eulerian graph.
3. Find an Euler tour, $\mathcal{T}$, of this graph.
4. Output the tour that visits vertices of G in order of their first appearance in $\mathcal{T}$. Let $\mathcal{C}$ be this tour.

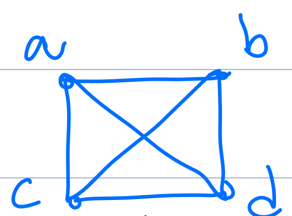
Remark:

Number of odd-degree vertices in a graph is even as sum of degrees of all vertices is even.

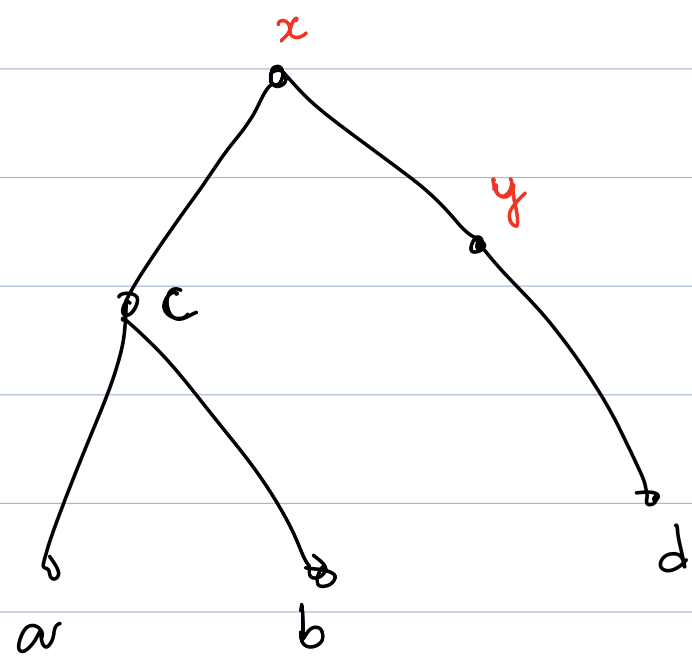
Idea:

$\{a, b, c, d\}$ are odd degree vertices.

Consider the subgraph induced by $\{a, b, c, d\}$



H



MST T

Find a ==perfect matching M of minimum weight== in H.

add the edges of M to T.

Analysis:

Let V' denote the set of odd degree vertices in MST T.

Lemma: Let $V' \subseteq V$, such that $|V'|$ is even and let M be a minimum cost perfect matching on V'. Then, $cost(M) \leq OPT/2$

Proof:-

Consider an optimal TSP tour of G, say L, let L' be the tour on V' obtained by short-cutting L. By the triangle inequality,

$$cost(L') \leq cost(L).$$

Observe, L' is the union of two perfect matchings on V', each consisting of alternate edges of L'.

Thus the cheaper of these matchings has cost $\leq cost(L')/2 \leq OPT/2$

Hence, the cost of minimum weight P.M is atmost OPT/2.

=

o/p of Algo

MST

matching

$$\text{Cost}(T) \le \text{Cost}(T) + \text{cost}(M)$$

$$\le OPT + OPT/2$$

$$= \frac{3}{2} OPT.$$

Note:

1. The above $\frac{3}{2}$-algorithm to metric TSP is the best known approximation algo as of now (recently some progress has been made)

2. Finding a better approximation algorithm for metric TSP (triangle inequality holds) is one of the outstanding open problems in this area.

* It is NP-hard to approximate metric TSP within a factor of $\frac{123}{122}$ [KLS15]

Q: What happens if we drop the assumption that cost function c satisfies the triangle inequality? Answer is given in the next page.

Theorem: If $P \neq NP$, then for any constant $\rho \geq 1$ there is no polynomial-time approximation algorithm with approximation ratio ρ for the general TSP.

ie, There is no constant factor approximation algorithm to TSP unless $P = NP$.

Proof:- Proof by contradiction,

Suppose that for some constant $\rho \geq 1$, there is a polynomial-time approximation algorithm A with approximation ratio ρ. WLOG, assume that ρ is an integer.

ie, $C(A) \leq \rho \, C(OPT)$, $\rho \geq 1$

We show how to use A to solve Hamiltonian-cycle problem in polynomial time. As Ham. cycle is NP-complete, we get $P = NP$.

Let $G=(V,E)$ be an instance of H. cycle Problem.

We build G' an instance of TSP from G as follows.

$G'=(V,E')$ be the Complete graph on V

ie, $E'=\{uv \mid u,v\in V,\ u\neq v\}$

Cost to each edge in E' is defined as

$$C_{uv}=\begin{cases} 1 & \text{if } uv\in E \\ \rho|V|+1 & \text{otherwise} \end{cases}$$

* c doesn't satisfy triangle inequality.

Clearly we can construct G' and c from G in Polynomial time.

Now consider the TSP (G',c).

Observe that

" If G has a Hamiltonian cycle then min cost TS Tour $=|V|$ "

If G doesn't have Hamiltonian cycle they min cost TS tour contain at least one new edge. So cost is

at least $\geq |V|-1 + \rho|V|+1$

$= \rho|V| + |V|$

$> \rho|V|$

Because edges not in G are so costly, there is a gap of at least $\rho|V|$ b/w the cost of a tour that is a hamiltonian cycle in G and the cost of any tour.

$\therefore$ the cost of a tour that is not a hamiltonian cycle in G is at least a factor of $\rho+1$ greater than the cost of a tour that is a Hamiltonian cycle in G.

Run the ρ-approximation algorithm A on G', if G has Hamiltonian cycle it will O/P a tour with cost $\leq \rho|V|$

otherwise A returns a tour of cost more than $\rho|V|$.

$\therefore$ we can use A to solve Hamiltonian cycle in Polynomial time.

Note that in the above reduction edge costs violate the triangle inequality.

In its full generality, TSP cannot be approximated assuming $P \neq NP$.