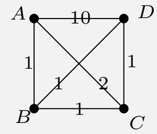
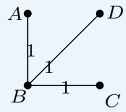
Problem 1. The figure below shows a weighted simple graph $G = (V, E)$, where the integers indicate the edge weights.



Explain why the approximation algorithm for TSP can no longer ensure an approximation ratio of 2 without triangle inequality.

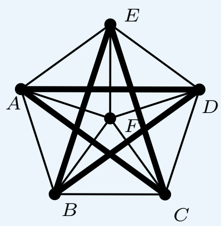
Solution.



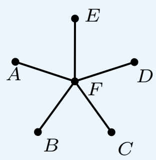
The above figure illustrates an MST of G. The algorithm generates a closed walk $A - B - C - B - D - B - A$, which gives us the Hamiltonian cycle $A \to B \to C \to D \to A$ with length 14. An optimal Hamiltonian cycle is 6 ($A \to B \to D \to C \to A$).

Problem 2. Show that the approximation algorithm for TSP does not guarantee an approximation ratio of 1.6.

Proof. Consider the graph with 6 vertices below, where the bolded edges carry the weights of 1, and the light edges carry the weights of 2.



An optimal solution is of the length 6, with the cycle $A - B - C - D - E - F - A$. Suppose that the algorithm finds an MST as shown below:



$FAFCFEFBFDF$ is a closed walk of this MST. Our algorithm will produce a Hamiltonian cycle $F \to A \to C \to E \to B \to D \to F$, whose length is 10. The approximation ratio is $10/6 > 1.6$. $\square$

Problem 3. Let $G = (V, E)$ be a simple undirected graph where each edge $e \in E$ is assigned a non-negative weight $w(e)$. G is connected. A *spanning walk* of G is a walk that visits every vertex at least once. Let OPT_G be the shortest length of all spanning walks. Design a poly($|V|$)-time algorithm to find a spanning walk with length at most $2 \cdot \mathrm{OPT}_G$.

Solution. Let λ be the weight sum of all edges in a minimum spanning tree of G. Any spanning walk must contain edges from some spanning tree. Hence, $\mathrm{OPT}_G \geq \lambda$.

Furthermore, the approximation algorithm for traveling salesman problem finds a spanning walk with length of at most 2λ, which is in turn at most $2 \cdot \mathrm{OPT}_G$.

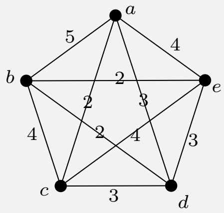
Problem 4. **(No Triangle Inequality No Approximation)** Prove: unless $\mathbf{P} = \mathbf{NP}$, it is not possible to guarantee any constant approximation ratio for the TSP problem in polynomial time, once the triangle inequality requirement is dropped.
Consider the **Hamiltonian Cycle Problem**, where the input is a simple undirected graph G (which may not be a complete graph) and it should output whether G contains a Hamiltonian cycle.
Reduce the Hamiltonian-cycle problem, which is an **NP**-Hard problem, to the TSP problem.

Proof. Suppose that we have an algorithm $\mathcal{A}$ for the TSP problem with approximation ratio $\rho \geq 1$ in polynomial time. Then, we can solve the Hamiltonian Cycle Problem in polynomial time using $\mathcal{A}$.

Given an input $G = (V, E)$, construct a complete graph $G' = (V', E')$ where $V = V'$ and for every edge $e = \{u, v\} \in E'$, set $w(e) = 1$ if $e \in E$, and $w(e) = \rho|V| + 1$. Run A on G' to find a Hamiltonian cycle C in polynomial time. Suppose that the length of C is $|V|$, then G has a Hamiltonian cycle. Note that C has exactly $|V|$ edges, then every edge of C found on G' must have weight 1, and hence exists in G. Also, if G has a Hamiltonian cycle, then C must have length $|V|$. As G has a Hamiltonian cycle, the cycle must also exist in G'. Therefore, the TSP problem on G' has an optimal solution whose length $\mathrm{OPT} = |V|$. If C uses any edge $e \in E'$ that does not exist in G. Then the length of C must be at least $(|V| - 1) + (\rho|V| + 1) = \rho|V| + |V|$, which is strictly larger than $\rho|V| = \rho\mathrm{OPT}$. This contradicts the fact that $\mathcal{A}$ is ρ-approximate.

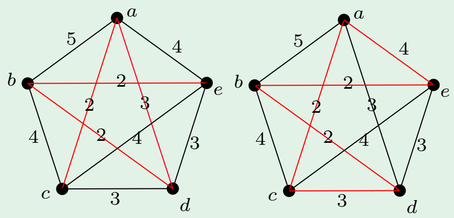
Hence, G has any Hamiltonian cycle if and only if the length of C is $|V|$. However, no polynomial time algorithm for the Hamiltonian-cycle problem exists, which implies that the no algorithm can guarantee any approximation ratio for TSP problem on a general graph in polynomial time. $\square$

Problem 5. Consider the undirected graph G below.



Use the algorithm taught in the class to find a Hamiltonian cycle that achieves an approximation ratio of 2.

Solution. The MST obtained has its edges colored in red, shown on the graph on the left; The graph on the right is the Hamiltonian cycle found by the algorithm, where the spanning walk obtained from the MST is *acadbebda* and the cycle obtained is *acdbea*.



Problem 6. Explain how to compute the walk in time proportional to the number of vertices in T by the MST computed.

Solution. Apply a depth-first search on T, which is done in $O(|V|+|E|) = O(|V|)$ time, where V is the set of vertices in T and E is the set of edges (with size $|V|-1$) in T.

Maintain a list of the vertices forming a walk, encountered by the depth first search. Append v to the end of the list whenever the DFS algorithm checks for a vertex v on top of the stack S. Since v must have its parent u underneath in the stack, u must be appended when v is popped, traversing back from v to u. Also, when v discovers a white out-neighbour u, u will be added to the stack and then to the list when DFS considers the next candidate on top of the stack, which is a traversal from v to u. Hence, the list indeed maintains a walk of T, and the algorithm also traverse through an edge twice by traversing in and out, which means every edge of T appears on the walk twice.

Problem 7. (Euclidean Traveling Salesman) Let P be a set of n points in 2D space. Define a cycle as a sequence of n line segments: $(s_1, t_1), (s_2, t_2), \ldots, (s_n, t_n)$ where

- $s_i \in P$ and $t_i \in P$ for each $i \in [1, n]$;
- $t_i = s_{i+1}$ for all $i \in [1, n-1]$ and $s_1 = t_n$;
- $P = s_1, s_2, \ldots, s_n$;
- each (s_i, t_i) is a segment connecting points s_i and t_i.

The length of the cycle is the total length of all the n segments. Let OPT_P be the shortest length of all cycles. Design a poly(n)-time algorithm (i.e., polynomial in n) that finds a cycle with length at most $2 \cdot \mathrm{OPT}_P$.

Solution. The length of a segment defined in the Euclidean space is the L_2-norm, satisfying the triangle inequality. Hence, we can cast the Euclidean Shortest Hamiltonian Cycle problem into the TSP problem by constructing a complete graph G.

Let $v_1, v_2, \ldots, v_n$ be the n points in P. For each pair of points (v_i, v_j) forming an edge in G, let the weight of this edge be the Euclidean distance between point i and point j.

Clearly, the shortest Hamiltonian cycle containing all the points in P is also a solution for the Traveling Salesman problem on G, meaning that $\mathrm{OPT}_G \leq \mathrm{OPT}_P$.

Then, obtain an Hamiltonian cycle with length at most $2 \cdot \mathrm{OPT}_G \leq 2 \cdot \mathrm{OPT}_P$ using the approximation algorithm for the Traveling Salesman problem, which is the desired.