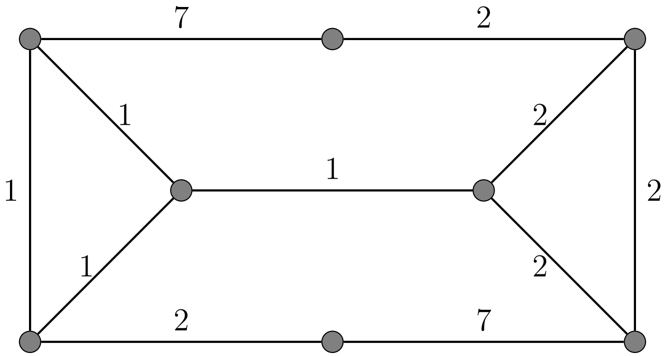
Introduction to approximation and randomized algorithms

2nd exercise session 30th October 2019

Exercise 1. Let us have the following graph:



a) What is the shortest Hamiltonian cycle in this graph?

b) What is the optimal solution of TSP in this graph?

c) What solution do we get from a run of Christofides algorithm?

Exercise 2. Prove that the algorithm for metric TSP that walks around the minimum spanning tree is not better than 2-approximation. That means construct a class of graphs $\{G_i | i \in \mathbb{N}\}$ such that $|V(G_{i+1})| > |V(G_i)|$ and for every $\varepsilon > 0$ there exists G_i the approximation ratio on this graph is worse than $2 - \varepsilon$.

Exercise 3. The traveling salesman problem is hard. But what if instead of cycle we would be looking for a more general structure? We are are going to look at problem of MINIMUM CYCLE COVER. On input we are given a directed graph $G = (V,E)$ with integral edge lengths ℓ_e and we want to find a subset of edges $F \subseteq E$ such that in graph (V,F) each vertex has an in and out degree equal to one and sum of lengths of edges in F is minimum. In other words, we want to cover vertices of G by disjunct directed cycles. Find a polynomial algorithm for this problem. There are at least two ways how to do it. If you know something from optimization and know what a total unimodularity is, then you can try to go that direction. Otherwise try to solve it by combinatorial construction. Let me just remind you that perfect matching of minimum cost can be found in polynomial time.

Exercise 4. Isn't it actually the case that the TSP can be solved in polynomial time? Consider the following algorithm based on dynamic programming. In the algorithm we will be constructing a table $d[i,u,v]$ for $i \in \{0,\dots,n\}$ and $u,v \in V$, where $d[i,u,v]$ will denote the length of path from u to v after i steps. We will denote the set of all neighbors of u by $N(u)$. The algorithm in given in a pseudocode:

$d[0,u,u] := 0\ \forall u \in V$, $d[0,u,v] := \infty\ \forall u \neq v \in V$
for $i = 1,\dots,n$ **do**
  for all $u,v \in V$ **do**
    $d[i,u,v] := \min_{w\in N(u)}(d(u,w) + d[i-1,w,v])$
    $d[i,v,u] := d[i,u,v]$

After the computation finishes the algorithm returns the minimal value $d[n,u,u]$ over $u \in V$. Does this algorithm solve TSP?

Exercise 5. Consider the following algorithm. We are going to construct a cycle for TSP in steps.

1. We'll start with a cycle C consisting of pair of vertices of minimum distance, that is pair of vertices u and v such that $d(u,v)$ is minimal.
2. Until all vertices are in the cycle C repeat the following two steps:
3. Choose vertex $u \in C$ and vertex $v \notin C$ such that $d(u,v)$ is minimal.
4. Add v into the cycle C right after u.

What does this algorithm construct?