

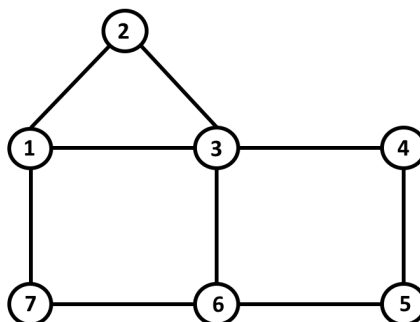
Note: Homework should be submitted in pairs on Canvas. We will only accept PDF files.

### 1. (Tree Width.)

- (a) Consider the following algorithm for constructing a tree decomposition of a given graph  $G$ : (1) Pick an arbitrary vertex  $v$  of  $G$ . (2) Remove  $v$  from the graph; for every pair of neighbors,  $u$  and  $w$ , of  $v$ , add an edge between  $u$  and  $w$ ; call the resulting graph  $G'$ . (3) Recursively find a tree-decomposition  $T'$  of  $G'$ . (4) Augment  $T'$  by adding at most one new node to it to obtain  $T$ .

Fill in the details for step (4) to show that it is always possible to construct a tree decomposition  $T$  of  $G$  from  $T'$  regardless of the choice of vertex  $v$ . Try to make the tree-width of your construction as small as possible.

- (b) Implement the algorithm on the graph below, removing nodes in the order  $1, 2, \dots$ . What is the tree-width of the tree decomposition you obtain? What is the minimum possible tree-width for this graph?



- (c) Prove that there is always an elimination ordering for the vertices of a given graph  $G$  that produces a tree decomposition of the minimum tree width.
2. **(Traveling Salesman Problem.)** We developed an approximation for the traveling salesman problem in class that relied on two lower bounds. Given a weighted complete graph  $G = (V, E)$ , let  $\text{MST}$  denote the weight of the minimum spanning tree in  $G$ . Let  $\text{MATCH}$  denote the maximum over all even sized subsets  $S$  of  $V$  of the weight of the min-cost perfect matching. Let  $\text{TSP}$  denote the weight of the optimal traveling salesman tour, and  $\text{ALG}$  denote the weight of the tour produced by Christofides' algorithm. We can think of  $\text{LB} = \max(\text{MST}, 2\text{MATCH})$  as a lower bound for  $\text{TSP}$ . We showed in class that  $\text{ALG} \leq \frac{3}{2}\text{LB} \leq \frac{3}{2}\text{TSP}$ .
- In each of the following parts you are asked to construct an example showing that some gap is at least  $\alpha$ . It is also acceptable to construct a family of examples where the gap can be made arbitrarily close to  $\alpha$ .
- (a) Construct a weighted graph  $G$  in which Christofides' algorithm obtains an approximation factor of  $3/2$ , showing that our analysis is tight.
- (b) Construct a (different) weighted graph  $G$  in which  $\text{LB}$  is equal to  $\frac{2}{3}\text{TSP}$ , showing that in order to perform better, we would need to use a better lower bound than  $\text{LB}$ .
3. **(Vertex Cover.)** Recall that a vertex cover of a graph  $G = (V, E)$  is a set of vertices  $S \subset V$  such that each edge has at least one endpoint in  $S$ . We analyzed the following algorithm in class and showed that it obtains a 2-approximation for the Vertex Cover problem.

- (i) Start with  $S \leftarrow \emptyset$ .
  - (ii) Pick an edge  $(u, v)$  such that  $\{u, v\} \cap S = \emptyset$ . Add both  $u$  and  $v$  to  $S$ .
  - (iii) If  $S$  is a vertex cover, halt, else go to Step (ii).
- (a) Consider changing step (ii) of the algorithm to the following: “Pick an edge  $(u, v)$  such that  $\{u, v\} \cap S = \emptyset$ . Add an arbitrary endpoint of the edge to  $S$ .”  
For any given integer  $n > 0$ , construct an instance of size  $n$  on which this algorithm may return a set  $\Omega(n)$  times larger than the smallest vertex cover.
  - (b) Next consider changing step (ii) to the following: “Pick an edge  $(u, v)$  such that  $\{u, v\} \cap S = \emptyset$ . Flip a coin and with probability  $1/2$  add  $u$  to  $S$ , else add  $v$  to  $S$ .”  
Show that this variant achieves a 2-approximation. (*Hint: Consider any vertex  $v$  in the optimal vertex cover, and let  $N(v)$  be the set of the neighbors of  $v$  as well as  $v$  itself. What is  $E[|S \cap N(v)|]$ ?*)
  - (c) Suppose each vertex  $v$  has a weight  $w(v)$ , and the objective is to pick a set of smallest weight. Give an example to show that the above algorithms (the original one and the two variants) do not work for this problem.
  - (d) Finally consider changing step (ii) so that upon picking edge  $(u, v)$ , we add  $u$  to  $S$  with probability  $\frac{w(v)}{w(u)+w(v)}$  and add  $v$  to  $S$  with probability  $\frac{w(u)}{w(u)+w(v)}$ . If  $W$  is the weight of the least-weight vertex cover, show that this variation obtains an expected weight  $E[\sum_{v \in S} w(v)] \leq 2W$ .
4. **(Linear Programming.)** In the  $s$ - $t$  shortest paths problem, you are given a directed graph with non-negative lengths  $\ell_e$  on edges, and special nodes  $s$  and  $t$ . Your goal is to find the length of the shortest path from  $s$  to  $t$ . Write down an LP to solve this problem exactly. Prove that the LP solves the problem exactly. The size of the LP should be polynomial in the size of the graph.