

Advanced Algorithms, Spring 2021

Homework 2. Due date: Thursday 29/4/21

Please follow the HW guidelines published in HW1

Question 1 (24 pts.)

Reduce the **Independent Set** optimization problem to the Independent Set decision problem. Directions: Assume that you have an algorithm A_d for the decision problem. Formally, $A_d(G,k)$ returns TRUE if G has an Independent Set of size k and FALSE otherwise. You need to describe an algorithm A_o that may run A_d a polynomial number of times. $A_o(G)$ should return a set of vertices forming a **maximal** Independent Set (MIS) in G . Divide the task into two stages, first find the size of the MIS, then the vertices forming it.

Question 2 (22 pts.)

Oren works in the HR department of a high-tech company. He organizes a big event with fun activities to the workers. There are m workers in the company, each of them selects her two favorite activities. Some of the activities will take place in the morning and some in the afternoon. Oren needs to partition the activities into the two shifts, morning and afternoon, in a way that maximizes the number of workers that can participate in their two favorite activities (one of their favorite activities is in the morning shift and the other is in the afternoon shift).

2.0. (warm-up, 0 pts.) Model the above problem as an optimization problem in an undirected and unweighted graph. The answer is below, skip it for now if you want to challenge yourself.

Given an undirected graph $G=(V,E)$, a **cut** is a bipartition $(S,V-S)$ of V . The size of a cut $(S,V-S)$ is the number of edges with one endpoint on each side of the cut. The **maximum cut** problem is that of finding a cut of maximum size.

For a set of vertices $A \subseteq V$ and a vertex $v \notin A$, let $c(v,A) = |\{vu \in E, u \in A\}|$, that is, the number of v 's neighbors that are in A . Consider the following algorithm for finding a cut:

1. Let v_1, v_2 be two arbitrary distinct vertices
2. Init $A \leftarrow \{v_1\}$, $B \leftarrow \{v_2\}$
3. Consider the vertices of $V - \{v_1, v_2\}$ in arbitrary order
For every considered $v \in V - \{v_1, v_2\}$:
If $c(v,A) \geq c(v,B)$ then $B \leftarrow B \cup \{v\}$ else $A \leftarrow A \cup \{v\}$
4. Return (A,B)

- 2.1.** (16 pts.) Prove that this is a 2-approximation algorithm for maximum cut.
- 2.2.** (6 pts.) Give the simplest example you can think of on which the algorithm returns a cut whose weight is exactly half the maximum cut.

Question 3 (22 pts.)

Let $G=(V,E)$ be an undirected graph with nonnegative edge costs. Let $R \subseteq V$ be a given set of **terminal** vertices. A **Steiner tree** is a tree in G that contains all the terminal vertices and any subset of the other vertices.

The Steiner tree problem is the problem of finding a minimum-cost Steiner tree.

The case $R=V$ is simply the Minimum Spanning Tree problem that, as you know, has an optimal polynomial time algorithm. However, the general problem is known to be NP-complete.

Given a **complete** graph G with weights that obey the triangle inequality, and a subset of vertices R , a possible algorithm for constructing a Steiner tree for R simply finds an MST on the subgraph induced by R (since the graph is complete this subgraph is also complete, and in particular is connected). Prove that this algorithm is a 2-approximation algorithm.

Guidelines: Let T^* be an optimal Steiner tree for R . Consider doubling each edge, and finding an Euler cycle P on the resulting graph (with parallel edges, as we did in class in the TSP approximation alg.). Next consider the path H that results from removing from P all vertices that do not belong to R and keeping only the first appearance of every vertex in R (shortcuts).

Question 4 (22 pts.)

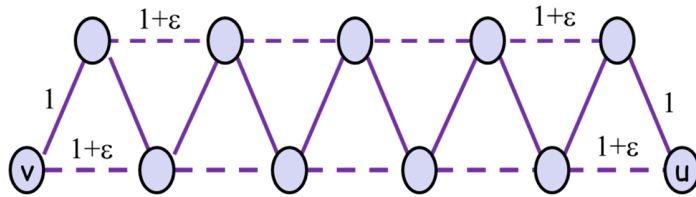
Given an instance of **Bin-Packing** in which all items have size at most $1/5$.

- 4.1.** (14 pts.) Show that Next-fit algorithm packs the items in less than $\frac{5}{4} \cdot \text{OPT} + \frac{3}{4}$ bins, where OPT is the minimal number of bins required to pack the items.
- 4.2.** (8 pts.) Show that the analysis is tight. Specifically, given n , describe an instance with $\Theta(n)$ items such that Next-fit uses at least $\frac{5}{4} \cdot \text{OPT} - \frac{1}{4}$ bins (which is one bin less than the bound provided by the analysis).

Question 5 (10 pts.)

Read and understand the following example, showing that the analysis of Christofides' Algorithm (3/2-approx to TSP) is tight. Consider the graph in the figure.

It consists of $2n+1$ nodes: n in the top row and $n+1$ in the lower row. All solid edges have weight 1, all dashed edges have weight $1+\epsilon$. The other edges are omitted from the figure, each has the maximal weight satisfying the triangle inequality (for example, (u,v) has cost $(1+\epsilon)n$).



Algorithm: The only MST consists of all solid lines, it is a path, and has length $2n$.

The first and last nodes have odd degree. The min-weight matching is simply the edge connecting them, which has length $(1+\epsilon)n$ (not drawn, by the Δ -ineq). All together, the TSP path detected by the algorithm has length $\sim 3n$.

An optimal tour: the dashed edges + the leftmost and rightmost edges.

Its length is $(2n-1)(1+\epsilon) + 2 \approx 2n$.

For full credit, declare that you read and understood the example and why it shows that the analysis is tight. For partial credit, specify what you failed to understand.