Problem 1

Consider the weighted set cover problem: we are given a universe $E = \{1, 2, ..., n\}$, subsets $S_1, S_2, ..., S_m \subseteq E$, and weights $w_1, w_2, ..., w_m$ for every one of the sets. Recall the analysis of the greedy algorithm for the unweighted set cover from the class:

Step 1: if E_i is the remaining number of elements at step i, then the set chosen in step i has size at least $\frac{E_i}{OPT}$;

Step 2: conclude that $E_{i+1} \leq E_i \cdot (1 - \frac{1}{OPT})$

Step 3: conclude that starting with $E_0 = n$ after $t = OPT \cdot \ln n$ steps we shall have $E_t = 0$, i.e., after taking $OPT \cdot \ln n$ sets we shall cover all the elements.

Perform a similar analysis for the weighted set cover problem. In the greedy algorithm for the weighted set cover, the algorithm chooses at each step j set S_{i_j} that maximizes

$$\frac{1}{w_{i_j}} \left| S_{i_j} \setminus \left(S_{i_1} \cup ... \cup S_{i_{j-1}} \right) \right| .$$

Start with proving the following inequality:

$$\frac{1}{w_{i_j}} \left| S_{i_j} \setminus \left(S_{i_1} \cup ... \cup S_{i_{j-1}} \right) \right| \geq \frac{E_j}{OPT}. \tag{1}$$

The above inequality corresponds to Step 1. Perform appropriate Steps 2 and 3, to show that also in the weighted case the greedy algorithms is a $\ln n$-approximation.

Hint: inequality $1 - x \leq e^{-x}$ might be handy.

Problem 2

In the unweighted partial set cover problem we are given a universe $E = \{1, 2, ..., n\}$, subsets $S_1, S_2, ..., S_m \subseteq E$, and a real parameter p from the interval $(0, 1)$. We need to find a sub-collection $S_{i_1}, S_{i_2}, ..., S_{i_t}$ such that

$$|S_{i_1} \cup S_{i_2} \cup ... \cup S_{i_t}| \geq p \cdot n$$

in a way that minimizes t (the number of sets). We again consider a greedy algorithm for the problem, which is exactly the same as the algorithm for the normal unweighted set cover. And again you need to follow the analysis that was recalled in Problem 1. In both, start with showing appropriate inequalities about the number of elements covered in one step (analogues of inequality (1)).

Task 1: Show that the greedy algorithm uses at most $\left(\ln \frac{1}{1-p} + 1\right) \cdot OPT_{SC}$ sets, where OPT_{SC} denotes the optimum number of sets that is needed to cover the whole universe E. I.e., OPT_{SC} is the optimum for the set cover problem on the instance $E = \{1, 2, ..., n\}$, subsets $S_1, S_2, ..., S_m \subseteq E$.

Task 2: Show that the greedy algorithm uses at most $(\ln(p \cdot n) + 1) \cdot OPT_{PSC}$, where this time OPT_{PSC} is the minimum number of sets that cover at least $p \cdot n$ elements of the universe.

Problem 3

Consider a graph $G = (V, E)$, and suppose that all vertices are numbered, i.e., $V = \{v_1, v_2, ..., v_n\}$. For a subset $S \subseteq V$ of nodes, let $\delta(S)$ be the number of edges $e = \{u, v\}$ such that $u \in S$ but $v \notin S$. In the problem of MAX-CUT we want to find a subset $S \subseteq V$ such that $\delta(S)$ is maximized.

In the class a randomized algorithm for the MAX-CUT was given. Here, we want to analyze its derandomized version:

1. $X_0 \leftarrow \emptyset$, $Y_0 \leftarrow \{v_1, ..., v_n\}$
2. **for** $i = 1$ **to** n **do**
   (a) $a_i = \delta(X_{i-1} + v_i) - \delta(X_{i-1})$
   (b) $b_i = \delta(Y_{i-1} - v_i) - \delta(Y_{i-1})$
   (c) **if** $a_i \geq b_i$ **then**
      i. $X_i \leftarrow X_{i-1} + v_i$ and $Y_i \leftarrow Y_{i-1}$
   (d) **else**
      i. $X_i \leftarrow X_{i-1}$ and $Y_i \leftarrow Y_{i-1} - v_i$
3. **return** X_n

Prove that the above algorithm is a 2-approximation, i.e., prove that

$$2 \cdot \delta(X_n) \geq \delta(OPT),$$

where OPT is the optimal set of nodes (maximizing $\delta(OPT)$). To prove the claim, follow the below steps:

Step 1: Consider sequence $S_i = \delta(X_i) + \delta(Y_i)$ for $i = 0, ..., n$. Show that the outcome of your algorithm is $\delta(X_n) = \frac{1}{2} S_n$. Hint: what is the relation between X_i and Y_i?

Step 2: Consider sequence $O_i = \delta\left((OPT \cup X_i) \cap Y_i\right) + \delta\left((\overline{OPT} \cup X_i) \cap Y_i\right)$, where $\overline{S} = V \setminus S$ just denotes the complement of a subset S. Show that $O_0 = 2 \cdot \delta(OPT)$, and that $O_n = 2 \cdot \delta(X_n)$.

Step 3: Show that function $S \mapsto \delta(S)$ is submodular, i.e., that for every two subsets S, T such that $S \subseteq T$ and element $v \notin T$ it holds that

$$\delta(T + v) - \delta(T) \leq \delta(S + v) - \delta(S).$$

Step 4: Using submodularity, show that the one-step increase of S_i is at least the one-step drop in O_i:

$$S_{i+1} - S_i \geq O_i - O_{i+1}.$$

Hint: to prove this inequality, it is convenient to consider two cases, when $v_{i+1} \in OPT$ and when $v_{i+1} \notin OPT$.

Step 5: From Steps 1,2,4 conclude that $2 \cdot \delta(X_n) \geq \delta(OPT)$.

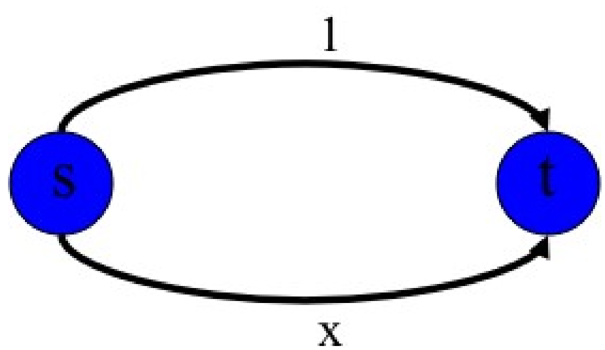
Problem 4 We are given a directed graph $G = (V, E)$ with edge weights c_e. We are given two nodes $s, t \in V$, and we want to find the shortest path between s and t. Consider the following relaxation of an integer program for the shortest $s - t$ path problem (here $\delta^{out}(u)$ denotes all arcs going out of u, and $\delta^{in}(u)$ denotes all arcs going into u):

$$\begin{array}{lcl}
\min & \sum_{uv \in E} c_{uv} x_{uv} & \\
\text{s.t.} & \sum_{uv \in \delta^{out}(u)} x_{uv} - \sum_{vu \in \delta^{in}(u)} x_{vu} = 0 & \forall_{u \neq s,t} \\
& \sum_{sv \in \delta^{out}(s)} x_{sv} - \sum_{vs \in \delta^{in}(s)} x_{vs} = 1 & \\
& \sum_{tv \in \delta^{out}(t)} x_{tv} - \sum_{vt \in \delta^{in}(t)} x_{vt} = -1 & \\
& x_e \geq 0 & \forall_{e \in E}.
\end{array}$$

Prove that there exists an optimal integer solution for the above program. Do so by writing a dual program, and construct an integer solution for the primal, and a solution for the dual, which will both have the same value. Construct the two solutions using Dijkstra's algorithm.

Problem 5

Suppose that we modify the Pigou's example in the figure so that the lower edge has cost function $c(c) = x^d$ for some $d \geq 1$. What is the price od anarchy of the resulting selfish routing network, as a function of d?



Problem 6 (Hard)

A 2-edge colorable graph is a graph in which we can color the edges with 2 colors, in a way that no edges of the same color share a vertex.

We are given a graph $G = (V, E)$ and we want to find a 2-edge colorable subgraph of G that has the maximum number of edges. Consider the following algorithm for the problem: find a maximum-size matching M_1 in G, and then find a maximum-size matching M_2 in $G - M_1$. Show that this algorithm is a $\frac{3}{2}$-approximation for the problem.