ALGORITHM DESIGN

Exercise Sheet 2

Lecturer: Prof. Stefano Leonardi
Teaching Assistants:
George Birmpas, Tomer Ezra

Academic Year 2022/2023
December 21, 2022
Due Date: 10 January 2023

General Terms: You can work on the exercises in teams of two people, and hand them in accordingly. However, make sure that each one of you has fully understood every solution you present. Do not copy any work from other students, the internet or other sources, and do not share your work with others outside your team. If at any point, any part of the exercises you hand in is apparent to be a copy of other work, this will result in the following consequences: All of your exercises, previous as well as upcoming ones, will be treated as if you did not hand them in at all, and you will have to participate in the written exam to make up for this. Please note that there will be no exception made, even if you are the original author of work someone else copied, or if your exercise partner is the one responsible. Therefore, please make sure to only choose a partner that you trust, and do not hand out your exercise solutions to others.

Late Policy: If you hand in your exercises after the due date, each day that you are late will result in a discount to your score, i.e. you will only receive 90% of the points if you are one day late, 80% on the second day after the due date, and 70% on the third. If you are late by more than three days, the assignment will get zero points in total.

Formal Requirements: Please typeset your solutions (11 pt at least), and state the names of both team members at the beginning of each sheet you hand in. Make sure to name the exact exercise each part of the solution refers to, otherwise it will not be graded. Please start a new page for each main exercise in the assignment (i.e. exercise 1, 2, etc., but not for each subquestion in them). Make sure your solution takes no more than one page for each main exercise (plus at most one extra page for the code, if an implementation is required). Everything after the first page will not be taken into account. So, for example, when the assignment has five exercises, please hand in five pages, one for each exercise. All solutions have to be sent via email to

birbas@diag.uniroma1.it

The subject of the email should be "Algorithm Design HW2-[Full-Name]", and the submitted file should be named "AlgorithmDesignHW2-[Full name].pdf".

Office Hours: There will be special hours for questions about the exercises announced via the piazza site. Presumably, this will take place in form of a zoom meeting .

Exercise 1

There is an underlying undirected graph $G = (V, E)$ with weights on the vertices, where the weight on vertex v is denoted by $w_v = d(v)$ (the degree of the vertex). Your goal is to find a vertex cover set U (a set where for every $e = (u, v)$, one of vertices u, v is in U) with the minimal total weight possible.

Show that any greedy algorithm, gives a 2-approximation to the best vertex cover.

Exercise 2

There are n arbitrary cards on the table each with a different number on it facing down (where n is an even number). You are allowed to flip the cards one by one and decide whether to select it or not before continuing to the next card. You are only allowed to select one card during this process.

1. Design a randomized strategy that guarantees that you select the card with the maximal value with a probability of at least $1/4$.
2. Prove the correctness of the guarantee of the strategy.

Exercise 3

Let $G = (V, E)$ be a **directed** graph, and let w_v be the weight of vertex v for every $v \in V$. We say that a directed edge $e = (u, v)$ is d-covered by a multi-set (a set that can contain elements more than one time) of vertices S if either u is in S at least once, or v is in S at least twice. The weight of a multi-set of vertices S is the sum of the weights of the vertices (where vertices that appear more than once, appear in the sum more than once).

1. Write an IP that finds the multi-set S that d-cover all edges, and minimizes the weight.
2. Write an LP that relaxes the IP.
3. Describe a rounding scheme that guarantees a 2-approximation to the best multi-set.

Exercise 4

Before we begin we will introduce the notion of Price of Stability (PoS). Let NE to be the set of all the equilibrium states, and s_{OPT} to be the state with the optimal Social Cost (SC). Then, the Price of Stability is defined as

$$PoS = \min_{s \in NE} \frac{SC(s)}{SC(s_{OPT})},$$

i.e., the ratio between the smallest Social Cost at an equilibrium state, and the optimal Social Cost.

Now, consider the following function

$$\Phi(\alpha) = \sum_{r \in R} \sum_{i=1}^{\#(r,\alpha)} c_r(i),$$

where α is a strategy profile $(\alpha_1, \ldots, \alpha_n)$, $r \in R$ is a resource, c_r is the cost function of resource r and $\#(r, \alpha)$ is the number of players using resource r in profile α. This is a potential function and it is called *Rosenthal's potential*.

Now consider a congestion game for which we have the following guarantee: The cost functions c_r are such that no resource is ever used by more than λ players. Use Rosenthal's potential to show that the Price of Stability of such a game is at most λ.

Hint: Try to use the observation that by starting from a state that produces the optimal social cost, in case this state is not a Nash equilibrium, then there is at least one player that by deviating can decrease the potential function.

Exercise 5

[Independent Set] There is an undirected graph $G = (V = \{v_1, \ldots, v_n\}, E)$. There are a total of $n!$ possible orderings of these vertices. Pick one such ordering uniformly at random $\sigma = (\sigma_1, \ldots, \sigma_n)$. Then consider the following process: Begin with $S = \emptyset$. Then, at each step (for $i = 1$ to n), if for all $u \in S$, $(u, v_{\sigma_i}) \notin E$, add v_{σ_i} to S.

Denote by d the maximum degree of a vertex of V. Prove that the proposed algorithm achieves an independent set with expected value of at least $1/d$ fraction of the optimal solution.