CSC 2420 Spring 2017, Assignment 2
Due date: March 27

It is certainly preferable for you to solve the questions without consulting a published source. However, if you are using a published source then you must specify the source and you should try to improve upon the presentation of the result.

If you would like to discuss any questions with someone else that is fine BUT at the end of any collaboration you must spend at least one hour playing video games or watching two periods of Maple Leaf hockey or maybe even start reading a good novel before writing anything down.

Unless stated otherwise, all subquestions are worth 10 points. If you do not know how to answer a question, state "I do not know how to answer this (sub) question" and you will receive 20% (i.e. 2 of 10 points) for doing so. You can receive partial credit for any reasonable attempt to answer a question BUT no credit for arguments that make no sense.

In class I can clarify any questions you may have about this assignment.

1. (a) Consider the following vertex weighted bipartite matching problem. We are given a bipartite graph $G = (X \cup Y, E)$, with $E \subseteq X \times Y$ and a positive integer capacity bound c_y for each $y \in Y$. The goal is to match vertices in X to vertices in Y such that for every $y \in Y$, no more than c_y vertices in X get matched to y. Use a max flow based algorithm to optimally maximize the size of the matching.

   (b) Consider the mackespan problem for the m machine restricted machines model. That is, every job $\mathcal{J}_i$ is represented by a pair (p_i, A_i) where p_i is the processing time (or load) for the job and $A_i \subseteq \{1, \ldots, m\}$ is the set of machines on which the job can be scheduled. Provide an optimal polynomial time algorithm for minimizing the makespane when all processing times identical (i.e $\forall i, p_i = c$ for some $c > 0$).

2. Consider the following *weighted partial vertex cover* problem: We are given a graph $G = (V, E)$ with node weights $c : V \to \mathbb{Q}^+$ and edge weights $d : E \to \mathbb{Q}^+$. The goal is to find a partial cover $V' \subseteq V$ so as to minimize the total cost of nodes in the cover V' plus the cost of edges not covered (i.e. edges not adjacent to at least one node in V') subject to the constraint that at most k edges are not covered.

   (a) Provide a {0,1} IP for this problem.

   (b) Using an LP relaxation and rounding of the IP, what is the approximation ratio that you obtain?

3. The following questions concern graph matching. For definiteness we can consider worst case complexity for deterministic algorithms but the following apply also to randomized algorithms (i.e. distributions over deterinistic algorithms) and also to stochastic analysis (i.e. distributions over inputs).

   (a) Let $\mathcal{A}$ be any algorithm that returns a *maximal* matching on some input graph instance G. Show that the size of the matching $|\mathcal{A}(G)| \geq \frac{1}{2}|OPT(G)|$ where OPT is any *maximum* matching for G. Thus any algorithm that always returns a maximal matching obtains (at least) approximation ratio $\frac{1}{2}$.

(b) Let G be a bipartite graph. Show that any online algorithm $\mathcal{A}$ can be converted to a *greedy* online algorithm $\mathcal{B}$ obtaining at least the same approximation ratio. Here greedy means that $\mathcal{B}$ will always match an online vertex if one of its neighbours is still available. Does anyone want a hint?

4. Consider the m machine makespan problem in the related machines model and suppose there are m_1 machines with speed 1 and m_2 machines with "slowdown" $s > 1$. We will say a job has *basic processing time* p if it takes time p to run on a machine at speed 1 and time $s \cdot p$ to run on a machine with slowdown s. Suppose all jobs have basic processing times in $\{1, 2, 3\}$

   (a) Show that it is an FPT problem (in parameter T) to determine if an instance has makespan T.

   (b) Show how to determine a solution if an instance has makespan T.

   Hint: It is an FPT problem (in parameter k) to solve an IP instance with k variables. You may use this fact.

5. We are given a degree bound $d << n$ and query access to a partial table for a function $f : Q \to Q$; namely given $\{(x_1, f(x_1), \dots, (x_n, f(x_n)\}$ we can access any $(x_i, f(x_i))$ in one query. Consider the following:

   (a) We want to test if the partial table f is produced by a degree d polynmial p or if it is "far-away" from any degree d polynomial where by far-away we mean that $f(x_i) = p(x_i)$ for at most $(1 - 2/d)n$ of the points given in the table. Provide a randomized 1-sided error algorithm that will make $O(d)$ queries, always returning p if it exists and with probability $\geq \delta$ will determine that f is far-away from any degree d polynomial. If neither condition is true, the algorithm can give any answer.

   Analyze the probability δ that can be achieved in terms of the number of queries used.

   (b) We now want to test if the partial table f is "close-to" a degree d polynomial p or "far-away" where far-away is as before and "close-to" means that $f(x_i) = p(x_i)$ for at least $(1 - 1/d)n$ of the points given in the table. Provide a randomized 2-sided error algorithm that will make $O(d)$ queries, returning p if it exists with probability $\geq \delta$ or determining with probability $\geq \delta$ that f is far-away from any degree d polynomial. If neither condition is true, the algorithm can give any answer.