CSC373S **Problem Set 3** **Winter, 2011**

Due: Monday, April 4, beginning of lecture

NOTE: Each problem set only counts 5% of your mark, but it is important to do your own work (but see below). As mentioned in class, this problem set has questions valued at 160 points but the assignemtn will only be graded out of 80 points so that this is an opportunity to raise your final grade by up to 5%. Similar questions may appear on the third term test as well as on the final exam. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. Anything else is *plagiarism*, and is subject to the University's Code of Behavior. You will receive 1/5 points for any (non bonus) question/subquestion for which you say "I do not know how to answer this question". You will receive .5/5 points if you just leave the question blank.

1. (20 points)

   Comment on the usefulness of the following two randomized algorithms. Namely, for each algorithm indicate (and justify in a couple of sentences) whether you can see some useful application for the algorithm.

   (a) INTEGER COMPOSITE TESTER: Given positive integer m,

   For $i := 1 \ldots, 100$
       Uniformly at random choose $r \in \{2, \ldots, \lfloor\sqrt{m}\rfloor\}$.
       If r divides m, then OUTPUT "m is COMPOSITE" and terminate
       EndIf
   EndFor
   Output "m is PRIME" and terminate

   (b) In the following , we use $*$ to denote polynomial multiplication and $\cdot$ to denote integer multiplication.

   POLYNOMIAL MULTIPLICATION TESTER: Given univariate polynomials $a(x), b(x), c(x)$ with integer coefficients and degrees (respectively) $m, m, 2m$:

   For $i := 1 \ldots, 100$
       Uniformly at random choose $r \in \{2, \ldots, m^2\}$.
       If $c(r) - a(r) \cdot b(r) \neq 0$, then OUTPUT $c(x) \neq a(x) * b(x)$ and terminate
       EndIf
   EndFor
   Output $c(x) = a(x) * b(x)$ and terminate.

2. (20 points)

   Consider the following weighted max-cut problem. The input is a graph $G = (V, E)$ with edge weights $w : E \to \Re^{\geq 0}$. A *cut* in the graph is a partition of the vertices; i.e. $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. The goal is to find a cut in the graph so as to maximize the weight of edges across the cut; that is, to partition the vertices so as to maximize $\sum_{e=(u,v):u\in V_1,v\in V_2} w_e$. Consider the following randomized algorithm:

$V_1 := \emptyset; V_2 := \emptyset$
For all $u \in V$
    With probability 1/2, $V_1 := V_1 \cup \{u\}$ else $V_2 := V_2 \cup \{u\}$
End For

Let RALG be the value of the cut produced by this algorithm. Show that $\mathcal{E}[RALG] = \frac{1}{2}\sum_{e\in E} w_e$.

3. (20 points)

   (a) Consider an exact 3-CNF formula F having 7 clauses. Show that F is satisfiable.

   (b) Consider the propositional formula $F = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee \bar{x}_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_4) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$

   Use the method of conditional expectations to compute a satisfying formula.

4. (20 points)

   Suppose we have a flow network $\mathcal{F} = (G, s, t, c)$ with integral capacities and suppose we are given an optimal integral flow f with $val(f) \geq 1$ in $\mathcal{F}$. Show how to efficiently find (say in time $O(|E|)$ an edge e in E whose capacity can be decreased by 1 unit so as to decrease the max flow to $val(f) - 1$. Briefly argue why reducing the capacity of this edge by one unit will reduce the max flow by one unit.

5. (20 points)

   Consider the following splittable makespan problem in the restricted machines model. We are given n jobs $J_1, \ldots, J_n$ and m machines where each job $J_i = (p_j, S_j)$ is described by a processing time (or load) p_j and an allowable subset of machines $S_j \subseteq \{1, \ldots, m\}$. The goal is to split the processing time of each job J_i on its allowable machines so as to minimize the maximum time (or load) that is assigned to any machine. Suppose all processing times are integral.

   Indicate how binary search and max flow can be used to obtain (in polynomial time) an optimal makespan bound L and splittable assignment of jobs to their allowable machines.

   Hint: Recall how max flow is used to solve the maximum bipartite matching problem.

6. (20 points)

   A graph $G = (V, E)$ is called $(k+1)$-clawfree if the neighbourhood $N(u)$ of every vertex u has at most k independent vertices. Here $N(u) = \{v \in V | (u, v) \in E\}$. Let $G = (V, E)$ be a $k+1$-clawfree graph with positive node weights; say w_u is the weight of node u and let $w(S) = \sum_{u\in S} w_u$ be the sum of weights in $S \subseteq V$. Consider the following local search algorithm for approximately finding a maximum weight independent subset in a $k+1$ clawfree graph $G=(V,E)$.

$S := \emptyset$
While $\exists u \in V - S$ such that $w_u > w(N(u) \cap S)$
    $S := S \cup \{u\} - N(S)$
End While

Show the locality gap (and hence the approximation ratio) for this local-search algorithm is at most k for every weighted $(k+1)$-clawfree graph.

7. (20 points) Consider the following *3 to 2 frequency set cover problem* : We are given a collection of sets $\mathcal{S} = \{S_1, \ldots, S_m\}$ for $S_i \subseteq U$ with the property that every $u \in U$ occurs in exactly three different sets S_i in $\mathcal{S}$. There is also a cost function $c : \mathcal{S} \rightarrow \Re^{\geq 0}$ and we let c_i denote the cost of set S_i. A feasible solution is a sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that every $u \in U$ occurs in at least two different sets S_i in $\mathcal{S}'$. The goal is to find a feasible solution $\mathcal{S}'$ so as to minimize the cost $c(\mathcal{S}') = \sum_{S_i \in \mathcal{S}'} c_i$.

    (a) Formulate the *3 to 2 frequency set cover problem* as a $\{0,1\}$ IP [10 points]

    (b) Show how to use LP relaxation + rounding to obtain an 2-approximation algorithm. Explain why your rounded solution is a feasible solution to the IP and why it provides a 2-approximation. [10 points]

8. (20 points)

    Consider the following call routing problem. There is an n node bi-directional ring network $G=(V,E)$ upon which calls must be routed. That is $V = \{0, 1, \ldots, n-1\}$ and $E = \{(i, i+1 \ mod \ n)\} \cup \{(i, i-1 \ mod \ n)\}$ and calls c_j are pairs (s_j, f_j) originating at node s_j and terminating at node f_j. Each call can be routed in a clockwise or counter-clockwise direction. The load L_e on any directed edge is the maximum number of calls routed on this edge. The goal is to minimize $max_{e \in E} L_e$.

    (a) Formulate this problem as an IP.
    Hint: Consider variables x_j and y_j that indicate the direction of call c_j. (You can also use just one indicator variable to represent the direction but I think it might be easier to think in terms of two such variables.)

    (b) Using an LP relaxation of this problem, show how to derive a 2-approximation algorithm.