Advanced Topics in Algorithms
Computer Science 182b
Spring 2011
Homework 3, Due Wednesday, February 9

1. [**20 Points**] **The Millisoft Matching Problem!** In this problem we investigate another discrete optimization problem and its corresponding matroid.

   Gill Bates, CEO of Millisoft, has come up with a revolutionary new scheme for matching up pen pals. He plans to offer the service through his MilliSoft Network web site in the near future.

   Assume that there are n subscribers to the service. Ideally, every subscriber in the network will be paired up with exactly one other subscriber. For compatibility reasons, only certain pairs of subscribers can be matched and it therefore may not be possible to find a match for every subscriber. Each subscriber stipulates a single fee that they are willing to pay to get matched. Millisoft would like to find a matching that maximizes the total fees that it can collect from the subscribers.

   This optimization problem can be modelled as a general graph in which vertices correspond to subscribers, edges correspond to subscribers that can potentially be matched, and the weight on each vertex represents the fee that will be paid if that vertex is matched. Our objective is to find a matching that maximizes the sum of the weights on the matched vertices.

   Show that this problem can be solved using the Matroid Greedy Algorithm by describing a matroid (and proving that it is indeed a matroid!) and showing that this problem reduces to finding a basis of maximum weight in that matroid. What is the total running time of your algorithm? Explain. (It needn't be the fastest possible algorithm, as long as it is polynomial time.)

2. [**15 Points**] **Chromatic Spanning Trees.** The Republic of Shmorbodia is planning to build a new spanning network to span all of its government buildings. Shmorbodia has k different private internet provides, each of which owns some set of the cables. Thus, you can imagine that each edge has a weight (cost) and a color, where the color indicates the provider that owns that cable. (The graph is simple; it is not a multigraph.) In the spirit of fairness, the Republic has set a limit m_i on the number of cables that can be rented from each provider i, $1 \leq i \leq k$. The objective is to find the spanning tree of minimum weight subject to the constraint on the number of cables rented from

each provider. Explain how this problem can be solved using an algorithm described in class.

3. **[30 Points] The Matroid Intersection Theorem.** The Matroid Intersection Theorem states that for any two matroids over the same ground set E, $M_1 = (E, I_1)$ and $M_2 = (E, I_2)$ with rank functions r_1 and r_2, respectively:

$$max_{X \in I_1 \cap I_2} |X| = min_{U \subseteq E}(r_1(U) + r_2(E - U))$$

   (a) Use this theorem to prove König's Matching Theorem that states that in a bipartite graph, the size of a maximum matching is equal to the size of a minimum vertex cover.

   (b) Use this theorem to prove " König's Other Theorem" (aka the König-Rado Theorem) that states that in a bipartite graph, the size of a minimum edge cover is equal to the size of a maximum independent set.

4. **[35 Points] The Dynamic List Access Problem!** In class we looked at the MTF online algorithm for the List Access Problem. Specifically, we looked at the **Static** List Access Problem where only FINDs were allowed - there were no INSERT and DELETE operations permitted in the request sequence.

   Now, consider the more general **Dynamic** List Access Problem which differs from the static version in three ways:

   (a) Any FIND operation may fail. That is, the item might not be found. In this case, the actual cost is $\ell + 1$ where ℓ was the length of the list at the time that the FIND was performed.

   (b) INSERT operations are permitted in the request sequence. An inserted object is placed at the end of the list. If the length of the list prior to the INSERT was ℓ, then the actual cost of this operation is $\ell + 1$. After inserting the element at cost $\ell + 1$, you may move it to any position earlier in the list at no cost (just like in a successful FIND operation).

   (c) DELETE operations are permitted in the request sequence. The cost of the DELETE is simply the cost of finding the object. (However, the potential function changes as a consequence of the DELETE, so a DELETE differs from a FIND in this way!)

   Describe a modification of the MTF algorithm to handle all three of these operations. Show that when your modified MTF algorithm is applied to the Dynamic List Access Problem, it still maintains a competitive ratio of 2. (That is, it is 2-competitive with respect to an optimal offline algorithm.)