CPSC 536N: Algorithms That Matter (Term 2, 2016-17)
Assignment 1
Due Friday March 3rd (negotiable)

Your solution must be written up in LaTeX. You may work in groups of at most 3.

Question 1: (A simple MapReduce problem)
The *prefix-sum* operator takes an array $a_1, \ldots, a_n$ and returns an array $s_1, \ldots, s_n$, where $s_i = \sum_{j \leq i} a_j$. For example, starting with an array $(17, 0, 5, 32)$, the output would be $(17, 17, 22, 54)$.

Describe how to implement prefix-sum in MapReduce, where the input is stored as $\langle i, a_i \rangle$. (So, the key is the position in the array, and the value is the array entry at that position.)

Question 2: Recall our discussion of maximizing a submodular function f subject to a matroid constraint:

$$\max \{\, f(S) \,:\, S \in \mathcal{I} \,\}$$

where $f : 2^E \to \mathbb{R}$ is non-negative, monotone, submodular and $M = (E, \mathcal{I})$ is a matroid. We showed that the "continuous greedy algorithm" and "pipage rounding" give a $(1 - 1/e)$-approximation (which is optimal for any efficient algorithm).

It is natural to wonder why we didn't just use the standard greedy algorithm instead. The pseudocode would be something like:

```
input f : 2^E → ℝ, matroid M = (E, I)
    Let r be the rank of the matroid
    Set S ← ∅
    for i = 1, ..., r
        Compute e ∈ arg max { f_S(e) : e ∈ E \ S,  S ∪ {e} ∈ I }
        Set S ← S ∪ {e}
```

Recall that $f_S(e)$ denotes the marginal gain $f(S \cup \{e\}) - f(S)$. It is known that this algorithm achieves a $1/2$-approximation.

Show that this algorithm actually does no better than a $1/2$-approximation! (If we were being a bit careful, we should say it does no better than $1/2 + \epsilon$ for any tiny $\epsilon > 0$.)

Hint: It suffices to consider a very small example with $|E| = 3$.

Question 3: Let $f : 2^E \to \mathbb{R}$ be any function (not necessarily submodular). Recall the definition of its multilinear extension:

$$F(x) = \sum_{S \subseteq E} f(S) \prod_{i \in S} x_i \prod_{j \in E \setminus S} (1 - x_j).$$

In class we discussed the fact:

$$\text{if } f \text{ is submodular} \qquad \Longrightarrow \qquad \frac{\partial^2 F}{\partial x_i \partial x_j}(x) \;\leq\; 0 \quad \forall i, j \in E,\ \forall x \in [0,1]^E.$$

In this exercise, you must prove the converse:

$$\frac{\partial^2 F}{\partial x_i \partial x_j}(x) \;\leq\; 0 \quad \forall i, j \in E,\ \forall x \in [0,1]^E \qquad \Longrightarrow \qquad \text{if } f \text{ is submodular}.$$

Question 4: In this problem we will think about how to maximize the transmission rate in a certain network with senders and receivers. Let $S = \{s_1, \ldots, s_a\}$ be the set of senders and $R = \{r_1, \ldots, r_b\}$ be the set of receivers.

Each receiver r_j has a demand α_j, which is the total rate that it desires to receive. Between s_i and r_j there is a network link which has a "critical capacity" $\eta_{i,j}$. The actual capacity $c_{i,j}$ is determined by a threshold τ_i chosen by sender s_i. The actual link capacity is:

$$c_{i,j} = \begin{cases} \tau_i & (\text{if } \tau_i \leq \eta_{i,j}) \\ 0 & (\text{otherwise}) \end{cases}.$$

We must also choose the rate $x_{i,j}$ at which s_i transmits to r_j. This must not exceed the link capacity, i.e.,

$$x_{i,j} \;\leq\; c_{i,j} \qquad \forall i, j.$$

The amount received by receiver r_j is

$$y_j \;=\; \min\Big(\sum_i x_{i,j},\, \alpha_j\Big).$$

The goal is to maximize $\sum_j y_j$, the total amount received by the receivers.

Suppose that each $\eta_{i,j} \in \{1, \ldots, M\}$. Design an algorithm with running time $\text{poly}(a, b, M)$ that achieves a $(1 - 1/e)$-approximation to the maximum transmission rate.

Hint: You may use the following fact without proof. Let $G = (V, A)$ be a directed graph with capacities on its arcs. Fix a "sink node" $t \in V$ and a set $S \subseteq V$ of "sources" with $t \notin S$. Define the function $f : 2^S \to \mathbb{R}$ where $f(U)$ is the maximum amount of flow that can be sent to t, using the nodes in U as sources. Then f is submodular.

Question 5: OPTIONAL BONUS QUESTION: Prove the hint from the previous question.