CS 271 (Spring 2013) — Assignment 1
Due: 01/24/2013

(1) Read Sections 2.1–2.4. No really, actually read them. You'll be amazed at how much easier a class gets when you actually read the chapters carefully as you go along. (You may also enjoy reading Sections 2.5 and 2.6. They will not be that essential for the class, but they're useful material.)

(2) Carefully read the course web site, `http://www-bcf.usc.edu/~dkempe/CS271/index.html`. Then answer the following questions, in your own words:

(a) On a homework not explicitly marked as "group homework", what sources of information are you allowed to use and what sources are you *not* allowed to use? To what extent can you collaborate with classmates?

(b) Describe the "Gilligan's Island" rule for homework in your own words.

(3) (a) Post (at least) one question on the Blackboard Discussion board. Ideally, this question would be one that you genuinely would like to know the answer to, and which relates to this class (either class policies, or the content that will be covered). If you cannot think of anything you don't know the answer to, feel free to find an interesting question about the material that you maybe do know the answer to.

(b) Post an answer on Blackboard Discussion to (at least) one question from a classmate. Your answer should show that you have given the question some thought; if the question was fairly difficult and your thoughtful answer happens to be wrong, that is not a problem. (Instead of an answer to a question, you can also post a correction to someone's incorrect answer.)

(4) Look for occurrences of two of our central class topics in the real world around you.

(a) Find an instance of something around you that can be (fairly) naturally modeled as a graph. (This should exclude social networks, the WWW, and the Internet, as they are a bit too obvious.) Find an interesting question — algorithmic or about the structure of that graph — that you figure it would be neat to know the answer to. Describe your related thoughts in about 1–2 paragraphs.

(b) Find an instance of something around you that either (1) you feel is naturally modeled by assuming randomness in a process, or (2) for which you think a *good* solution would involve doing things randomly. Give a brief (1–2 paragraphs) description for why you think randomness here could be natural or useful.

(Note: the corresponding chapters of the textbook almost certainly contain suitable answers to these questions; here, we want you to actually use your own eyes and creativity a bit instead.)

(5) Solve the following exercises from the textbook

(a) Section 2.1, Exercises 10, 18, 28, 44

(b) Section 2.2, Exercises 2, 50

(c) Section 2.3, Exercises 4, 14, 16, 36

(d) Section 2.4, Exercises 6, 16, 22, 32, 44

(6) [0 points]
Chocolate Problem (3 chocolate bars): Suppose that you are in a specialty chocolate store. The store produces n different types of chocolate; let's say they are numbered from $1, \ldots, n$. They are bundled into m different types of gift baskets. Gift baskets of type $i = 1, \ldots, m$ can be thought of as

sets $S_i \subseteq \{1, \ldots, n\}$: each basket of type i contains exactly one sample of each chocolate type j with $j \in S_i$. Each gift basket costs the same amount of money, and unfortunately, you only have money to buy $k < m$ baskets. But you want to sample as many types of chocolate as possible. (Having multiple samples of the same type does not add any value for you.) If you pick a set $I \subseteq \{1, \ldots, m\}$ of baskets, you get to sample all the chocolates in $\bigcup_{i \in I} S_i$. So you want to find a set I with $|I| \leq k$ such that $|\bigcup_{i \in I} S_i|$ is as large as possible.

One way to try to do this is the following: Start by picking the basket i with the largest number of different types of chocolate. Next, pick the basket i' that gives you the largest number of types that you don't have in your first basket. Now, pick i'' that gives you the largest number of new types you don't have in either i or i' so far. Continue this way until you have k baskets.

(a) Give an example where the algorithm above does not actually maximize the number of types of chocolate you get to try. (This is just meant as a warmup to get a feel for the problem. By itself, it's not worth any chocolate yet.)

(b) Let I^* be the actual best collection of baskets, and let $A^* = |\bigcup_{i \in I^*} S_i|$ be the number of types of chocolate it gets. Let $A = |\bigcup_{i \in I} S_i|$ be the number of types of chocolate the algorithm gets. Prove that $A \geq (1 - 1/e) \cdot A^*$, where $e \approx 2.71828 \cdots$ is Euler's constant. (So the algorithm actually comes pretty close to the best possible number of chocolate types.)

Here are a few hints: For each iteration, think about how well the algorithm could do by adding the best set from I^*. You might get a recurrence for the number of chocolate types you have after any number $k' \leq k$ steps. Solve that recurrence. You may want to remember what $(1 - 1/k)^k$ converges to, and from which side.