CS5330: Randomized Algorithms

Problem Set 1

Due: January 22, 6:30pm

Instructions. The *exercises* at the beginning of the problem set do not have to be submitted—though you may. The first three exercises should be easy review of basic probability (i.e., material you already know prior to this class). The remaining two exercises are more interesting and involve the minimum cut algorithm from class. There are three problems to submit. The first involves a basic probability calculation, while the second is a simple algorithm. The last question is a bit more involved and requires generalizing the algorithm and analysis from class.

- Please submit the problem set on IVLE in the appropriate folder. (Typing the solution using latex is recommended.) If you want to do the problem set by hand, please submit it at the beginning of class.
- Start each problem on a separate page.
- If you submit the problem set on paper, make sure your name is on each sheet of paper (and legible).
- If you submit the problem set on paper, staple the pages together.

Remember, that when a question asks for an algorithm, you should:

- First, give an overview of your answer. Think of this as the executive summary.
- Second, describe your algorithm in English, giving pseudocode if helpful.
- Third, give an example showing how your algorithm works. Draw a picture.

You may then give a proof of correctness, or explanation, of why your algorithm is correct, an analysis of the running time, and/or an analysis of the approximation ratio, depending on what the question is asking for.

Advice. Start the problem set early—questions may take time to think about. Come talk to me about the questions. Talk to other students about the problems.

Collaboration Policy. The submitted solution must be your own unique work. You may discuss your high-level approach and strategy with others, but you must then: (i) destroy any notes; (ii) spend 30 minutes on facebook or some other non-technical activity; (iii) write up the solution on your own; (iv) list all your collaborators. Similarly, you may use the internet to learn basic material, but do not search for answers to the problem set questions. You may not use any solutions that you find elsewhere, e.g. on the internet. Any similarity to other students' submissions will be treated as cheating.

Easy Exercises and Review (*May not be submitted.*)

Exercise 1. A fair coin is one that comes up heads with probability $p = 1/2$ and tails with probability $(1 - p) = 1/2$. Flip a fair coin ten times. What is the probability that:

a. The number of heads is equal to the number of tails?

b. There are more heads than tails?

Exercise 2. For each of the following, explain your answer:

a. Suppose that you roll a fair die that has six sides numbered $1, 2, \ldots, 6$. Is the event that the number on top is a multiple of 2 independent of the event that the number of top is a multiple of 3?

b. Suppose that you roll a fair die that has four sides numbered $1, 2, 3, 4$. Is the event that the number on top is a multiple of 2 independent of the event that the number of top is a multiple of 3?

Exercise 3. The FastCut algorithm reduces the problem from size n to $n/\sqrt{2} + 1$ and recurses twice on the remaining graph. Consider a version that reduces the problem from size $n/\sqrt{2}$ and recurses three times on the remaining graph. This would yield the following recurrence:

$$T(n) = 3T(n/\sqrt{2}) + n^2$$

Solve this recurrence.

More Interesting Exercises (*May not be submitted.*)

Exercise 4. Recall in class we described the following algorithm for finding a min-cut in a graph $G = (V, E)$:

- Assign each edge in the graph G a weight randomly chosen from $[0, 1]$. (If the weights are not unique, repeat.)
- Find an MST T of the weighted graph G.
- Let e be the maximum weight edge in T. By deleting edge e from the MST, it divides the tree into two sets of nodes: V_1 and V_2. Return these two sets as your cut.

Answer the following two questions:

a. Explain why this algorithm finds a min-cut with probability at least $2/(n \cdot (n-1))$.

b. Modify this algorithm using the ideas from the FastCut algorithm so that it finds a min-cut with probability at least $1/\log n$. (Hint: you will not want to simply run the MST algorithm to completion, but instead stop early and *do something*.)

Exercise 5. Assume that graph $G = (V, E)$ is a weighted graph. Give an algorithm for finding a min-cut in graph G with probability at least $1/(n \cdot (n-1))$. (Hint: Think about choosing each edge with proportion to its weight.) Prove that your algorithm is correct.

Problem 1. [*ESP*]

Dr. P. T. Fogg[1] is a noted research in telepathy. In order to prove the existence of telepathy, he runs the following experiment: Two psychics are positioned on opposite sides of an opaque, sound-proof barrier. Each has a fair six-sided die. Each time the test is run, the two psychics each roll their own die, and then attempt to guess the value of the *other* psychic's die. If telepathy does not exist, each psychic has a 1/6 chance of guessing the correct value.

Dr. Fogg ran his experiment using the following policy for determining how many times the test is to be run: the psychics rolls their dice repeatedly until both roll a six. At that point, the experiment is immediately halted, before the psychics can make a guess.

Problem 1.a. Surprisingly, Dr. Fogg finds that the psychics are guessing correctly with a probability slightly better than 1/6! Explain, in words, the flaw in this experimental design.

Problem 1.b. If a psychic exploits this flow optimally, with what probability can she guess the number on the opposite die?

[1]The "T" stands for "telepath."

Problem 2. [*All minimum cuts.*]

Give a (simple) algorithm for finding *all* the min-cuts in a graph, with high probability. Prove that your algorithm is correct. (Hint: if your proof involves a long analysis of graph structure, then there is a simpler solution.)

Problem 3. [*Almost minimum cuts.*]

Given a graph G with a min-cut of size k, let C be a cut of size (exactly) $\alpha \cdot k$. (For simplicity, you may assume that 2α is an integer, though it is not necessary for the final conclusion.)

Problem 3.a. Assume you run $Collapse(G, n, 2\alpha)$, i.e., you repeatedly contract edges until there are only 2α nodes left. Show that the probability that no edge in C was contracted is at least $\Omega(1/\binom{n}{2\alpha})$.

Problem 3.b. Assume you have a graph with 2α nodes containing cut C. Choose a cut at random from the graph. What is the probability that you return graph cut C?

Problem 3.c. Prove that there are at most $O((2n)^{2\alpha})$ cuts of size (exactly) $\alpha \cdot k$.

Problem 3.d. (*Challenge:*) Assume that for any (real) value of α, there are at most $O((2n)^{2\alpha})$ cuts of size $\alpha \cdot k$. Let G be a connected graph with a minimum cut size of k.

Mark each edge in G with probability $c \ln(2n)/k$, for some constant c. (Note that this is only meaninful if $k > c \ln(2n)$.) Delete every unmarked edges from the graph. Prove that with high probability graph G is still connected.

(The claim is true with high probability of the probability is at least $1 - 1/n^{c'}$, where given c' we can choose c above to yield the proper result. For example, we might choose $c = c' + 8$.)

Hint: a graph is connected if, for every cut, there is at least one edge across the cut.