15-859(D) Randomized Algorithms
Notes for 9/29/98

```
* randomized rounding
* probabilistic method
  - method of conditional expecations
```

Randomized routing/rounding

Given an undirected graph and a set of pairs $\{(s_i, t_i)\}$ we want to route these pairs to minimize the maximum congestion. This problem is NP-hard. Can we find an approximate solution?

Idea: (Raghavan & Thompson)

1. Solve fractionally. Think of as multi-commodity flow (e.g., allow s_i to route to t_i by sending 1/2 down one path, 1/4 down another path, and 1/4 down another). Can solve with linear programming: for each (directed) edge e, and each commodity i, have variable X_{ei}. Constraints for inflow = outflow. Constraints $\forall e, \sum_i X_{ie} \leq C$, and minimize C.

2. For each pair (s_i, t_i) we have a flow. Now what we do is view these fractional values as probabilities and select a path such that the probability we pick edge e is equal to the flow of this commodity on e. How can we do this algorithmically? (Give proof that greedy approach works.)

Analysis: fix some edge. Let f_i be the flow of commodity i on this edge. This also means that f_i is the probability that we picked this edge for routing (s_i, t_i). So, for a given edge, can think of $\{0, 1\}$ random variables X_i corresponding to event that we picked this edge for commodity i, where $\mathbf{Pr}[X_i = 1] = f_i$. For a given edge, these X_i are all INDEPENDENT. (Not independent for the same i across different edges, but that's OK). Expected value of sum is at most C. Now apply Chernoff.

$$\mathbf{Pr}[total > (1 + \epsilon)C] < e^{-\epsilon^2 C/3}$$

The point now is if this is small enough (e.g., $o(1/n^2)$) then the probability that *there exists* an edge whose congestion exceeds this bound is also small ($o(1)$).
So, if $C \gg \log(n)$, then w.h.p., maximum is only $1 + \epsilon$ times larger than the expectation.
What if $C = 1$, or C is constant? In this case, we can apply the bound:

$$Pr[total > kC] < (e^{k-1}/k^k)^C$$

So, set k to be $O(\log(n)/\log\log(n))$, and then get $1/poly(n)$.

The probabilistic method

Spencer's "graduation/tenure game":

- Begin with some set of people at various positions (number of years to go to graduate).
- 2 players: Paul (partitioner) splits people into two groups. Carole (chooser) selects one group to be removed, other advances.
- Start: people at various years until graduation. Paul wins if somebody graduates, Carole wins if nobody graduates.

Question: given a starting configuration, who wins? Note: this is a full information 2-player game. So, for a given starting configuration, somebody has a winning strategy. Let's say that at the start, there are a_k people at distance k from graduation (call this "level k").

Theorem 1: If $\sum_k a_k/2^k < 1$, then Carole has a winning strategy.
Theorem 2: If $\sum_k a_k/2^k \geq 1$, then Paul has a winning strategy.

Proof of Theorem 1. Suppose Carole plays randomly. What is the expected number of people that reach the goal? For instance, for fixed person who is initially at level k, what is the probability that this person reaches the goal? Therefore, no matter what Paul's strategy is, a random Carole has a non-zero chance of winning. Therefore, Paul cannot have a winning strategy and so Carole does. ∎

> **"The Probabilistic Method"**: We want to show that something exists. Do this by setting up some probability distribution and showing that what we want happens with nonzero probability. E.g., like when we showed that RP $\subseteq$ P/poly.

Actually, we can figure out Carole's winning strategy. Ideas? Just use Theorem 1: pick subset where sum is less than 1/2, so when they advance, the sum is less than 1, and we still have a winning strategy.

> **"Method of conditional expectations"**: General idea: say we've shown that making random choices results in a good expected value. Now we consider our first choice, and calculate conditional expectations given this first choice, and we pick one where conditional expectation is still good (which we know must occur, for appropriate definitions of "good").

Proof of Theorem 2: Suppose Paul can split so that both groups have sum $\geq 1/2$. Then the preconditions of the theorem are maintained and Paul wins. How can Paul do this? (Go through the elements from smallest k to largest k.) ∎

[Note: similar game: 20 questions with k lies allowed. questioner splits the set, and the chooser gives answer. Those that do not agree with answer move up one. position of element corresponds to the number of lies if element is the correct one. questioner's goal is to reduce the number of elements below the k-boundary to 1 as quickly as possible.]