

## Class 2: Agenda, Questions, and Links

### 1 Welcome Back!

Glad you came back! Let's quickly check in. How are you feeling about the mini-lecture on Karger's algorithm? Go to <http://PollEv.com/cs265> to answer.

(Frowny face = "didn't watch it"; Smiley face = "pretty happy with this material.")

### 2 Announcements

- Technical material from last class is now available on Canvas in a new video.
- Class "Meet-and-Greet" event, **this evening** (Wednesday 9/16), 5-6pm Pacific time, on Nooks. No agenda (other than maybe some icebreakers...), just a chance to get to know each other better.
- More information about homework!
  - First HW will be out soon (by Friday); due Friday 9/25 on Gradescope.
  - See course website for info about preparing/submitting HW.
  - HW will be done in small groups (up to three). It's okay to collaborate between groups (within reason—don't form study groups of 50 people), but each group should do its own write-up.
  - Each group should turn in exactly one homework assignment.
  - In addition to office hours and Piazza, we will have "Homework Parties" on Monday evenings, 6-7pm. At least one CA will be around, and it's a good time to meet up to start thinking about the homework.
- How do we find groups?
  - If you've been working with people in class and that's been going well, consider forming a HW group.
  - There's a "find teammates" post/feature on Piazza.
  - Hang out after class today and look for groups.
  - Come to the Meet-and-greet this evening to find people to work with.
  - If by **Friday**, you haven't found a group but you want to find a group, you can let us know and we will suggest (randomized) groups; we will send out a form on Friday for you to fill out.

- How do we work in groups?
  - Everyone should take an active role in brainstorming solutions, and equitably split up the writing. (For example, each person writes up some of the solutions. Or if you plan to stay with the same group for multiple weeks, rotate week-to-week who does the writing).
  - Schedule time(s) to “meet” outside of class to work on problems together. You can meet during our Monday “Homework Parties,” or come to our Nooks community any time! (Or whatever other platform you prefer).

### 3 Questions?

Any questions from the short lecture videos?

- Break into groups, and ask any questions that you have of each other (5 minutes).
- Put any questions that the group can’t resolve in the chat!

### 4 Coupon Collecting

This will be a good chance to practice the power of linearity of expectation.

(One-slide presentation to set up the problem; summary below)

Let  $W$  be a collection of  $n$  words. For example,  $W = \{\text{hello, randomized, algorithms, penguin, spaceship, \dots, mushroom}\}$ . There’s a button in front of you. Each time you push the button, it will say a word uniformly at random from  $W$ . If you push the button multiple times, it answers independently each time. (In particular, it’s possible to get the same word twice). The question is:

What is the expected number of times you push the button before you see all  $n$  words?

#### 4.1 Group work

You’ll answer this question by answering the following questions in your groups.

Let  $X_i$  be the time at which you see the  $i$ ’th new word. For example, if you push the button six times and see

penguin, penguin, randomized, hello, randomized, spaceship

then  $X_1 = 1$ , since you saw your first new word (“penguin”) on push 1.  $X_2 = 3$ , since you saw the second new word (“randomized”) on push 3. And  $X_3 = 4, X_4 = 6$ .

## Group Work

**Important:** as you make progress on the question(s), one person in each room should record your progress on <http://PollEv.com/cs265>.

First, introduce yourselves to each other. Then, start thinking about the following questions:

1. What is  $\mathbb{E}X_1$ ? (This is not a trick question).
2. What is  $\mathbb{E}(X_2 - X_1)$ ? That is, in expectation, how many times do you press the button, after you have seen the first word, before you see a new, second word?
3. What is  $\mathbb{E}(X_3 - X_2)$ ?
4. For any  $i = 2, 3, \dots, n$ , what is  $\mathbb{E}(X_i - X_{i-1})$ ?
5. Use your answers to the above, plus linearity of expectation, to answer our question: what is the expected number of times you push the button before you see all  $n$  words? It's okay if your answer is a summation, but if you have time try to simplify it to get a big-Theta expression.

## 4.2 Discussion

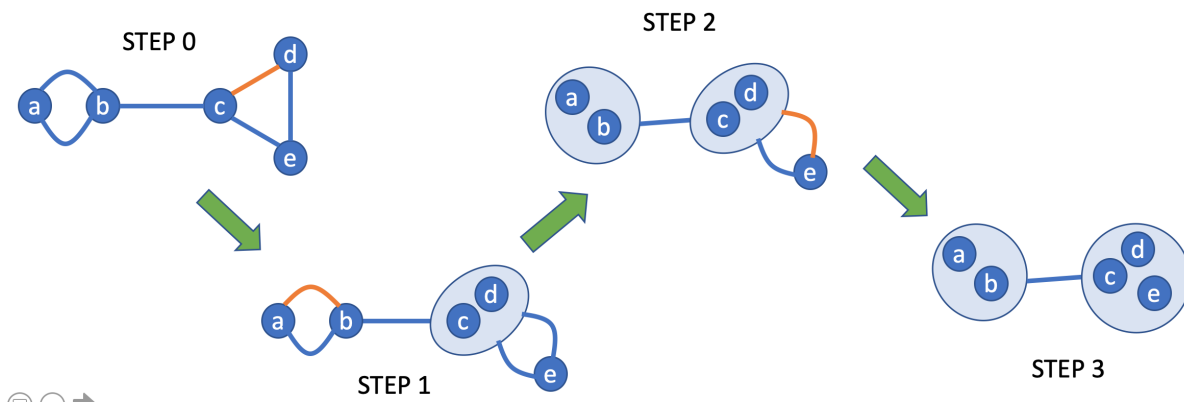
(A few slides to go over the solutions).

## 5 Karger-Stein Algorithm

Next, we'll take a look at a way to speed up Karger's algorithm.

### 5.1 Motivation for Karger-Stein

Here is one small run of Karger's algorithm:



## Group Work

**Important:** as you make progress on the question(s), one person in each room should record your progress on <http://PollEv.com/cs265>.

What is the probability of failure at each point in this run? That is, what is the probability that we choose an edge crossing the minimum cut?

(Slides: solutions, idea to do better, and an algorithm “Modified-Karger”)

Summary: Repeating Karger’s algorithm is wasteful since earlier steps are more likely to be successful than later steps; we should repeat later steps more.

MODIFIED-KARGER:

1. Start with a graph  $G$  on  $n$  vertices.
2. Run Karger’s algorithm (once) until there are  $m$  vertices remaining. Call the graph you end up with  $G'$ . (Note that  $G'$  will have some “mega-vertices” comprised of merged vertices).
3. Repeat Karger’s algorithm  $k$  times independently on  $G'$  until we end up with only two mega-vertices. Return the smallest cut we find.

## 5.2 Group Work

Consider the Modified-Karger algorithm above. In this group work, you will analyze this algorithm.

### Group Work

**Important:** as you make progress on the question(s), one person in each room should record your progress on <http://PollEv.com/cs265>.

1. Give a bound on the probability, in terms of  $n$  and  $m$  and  $k$ , that MODIFIED-KARGER is successful. (You may not be able to find the probability exactly, but give a decent lower bound, like we did for the original Karger’s algorithm; it’s okay if your expression is a bit complicated).

You may want to consult the lecture notes on Karger’s algorithm. They are available on the course website: [cs265.stanford.edu](http://cs265.stanford.edu)

**Hint:** You can break up the failure probability into two parts: the probability that we choose an edge crossing the mincut when reducing  $G$  to  $G'$ , and the probability that we choose an edge crossing the cut in *all* of the  $k$  runs of Karger of  $G'$ .

**When you finish with this, or if you get stuck, register that at [pollev.com/cs265](http://pollev.com/cs265)**

2. Choose  $m = \sqrt{n}$  and  $k = n \log n$ . Use part 1 to show that the success probability of MODIFIED-KARGER is  $\Omega(1/n)$ .

**Hint:** The fact that we're looking for a  $\Omega(\cdot)$  answer means that it's okay to ignore pesky constants in your analysis.

**Hint:** You might want to use the useful fact that  $1 - x \leq e^{-x}$  for all  $x$ .

**When you finish with this, or if you get stuck, register that at [pollev.com/cs265](http://pollev.com/cs265)**

3. Show that if you repeat MODIFIED-KARGER, with the parameters above,  $\Theta(n)$  times, you can obtain a success probability of 0.99.

**When you finish with this, or if you get stuck, register that at [pollev.com/cs265](http://pollev.com/cs265)**

4. How does this compare to the original Karger's algorithm? More precisely:

We saw in the mini-lecture that repeating Karger's algorithm  $O(n^2)$  times leads to a success probability of 0.99. This would involve  $O(n^3)$  edge contractions ( $n$  edge contractions per run of Karger's algorithm).

You've just shown that "repeating MODIFIED-KARGER  $\Theta(n)$  times" also has a success probability of 0.99. How many edge contractions does this involve?

**When you finish with this, or if you get stuck, register that at [pollev.com/cs265](http://pollev.com/cs265)**

5. If you're done with the above and we haven't finished group work yet, you can think about the following challenge problem:

Above, we saw how to improve Karger's algorithm by breaking one step down into two steps. What if we were to iterate on this idea? (That is, instead of just running Karger on  $G'$ , recursively run MODIFIED-KARGER on  $G'$ ). How would you pick the parameters here? How few edge contractions can you get, if you want success probability 0.999?

## 5.3 Discussion

(Slides with solutions to questions 1-4)

If time, we'll discuss one potential solution to question 5, or hear your ideas!