CS265, Winter 2021-2022

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? (Use the OhYay emoji or text reactions).

2 Announcements

- First homework up now!
- Enrollment issue is fixed maybe? Can anyone confirm or deny?
- There should now be a "Question Board" in each breakout room. These are shared between the rooms, and the staff will see them. You can ask questions, or upvote others' questions!

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, $W = \{$`hello`, `randomized`, `algorithms`, `penguin`, `spaceship`, …, `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

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 $i = 2, 3, \ldots, n$, 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.

Group Work: Solutions

1. $\mathbb{E}X_1 = 1$.
2. $\mathbb{E}(X_2 - X_1) = \frac{1}{1-1/n}$. This is because we have a $1 - 1/n$ chance of getting a word other than the first one. We saw in the lecture video on linearity of expectation (or you know anyway) that the expected number of times you have to flip a p-biased coin before seeing heads is $1/p$, so the answer is $1/(1 - 1/n)$.
3. $\mathbb{E}(X_3 - X_2) = \frac{1}{1-2/n}$.
4. $\mathbb{E}(X_i - X_{i-1}) = \frac{1}{1-(i-1)/n}$

5. We have

$$\begin{aligned}\mathbb{E}X_n = \mathbb{E}X_1 + (X_2 - X_1) + (X_3 - X_2) + \cdots + (X_n - X_{n-1}) &= \mathbb{E}X_1 + \sum_{i=1}^{n-1} \mathbb{E}(X_{i+1} - X_i) \\ &= 1 + \sum_{i=1}^{n-1} \frac{1}{1 - i/n} \\ &= 1 + \sum_{i=1}^{n-1} \frac{n}{i} \\ &= 1 + n \sum_{i=1}^{n-1} \frac{1}{i} \\ &= \Theta(n \log n).\end{aligned}$$

To see this last thing, you could approximate $\sum_{i=1}^{n} \frac{1}{i} \approx \int_{x=1}^{n} \frac{1}{x}\, dx = \log(n)$.

4.2 Discussion

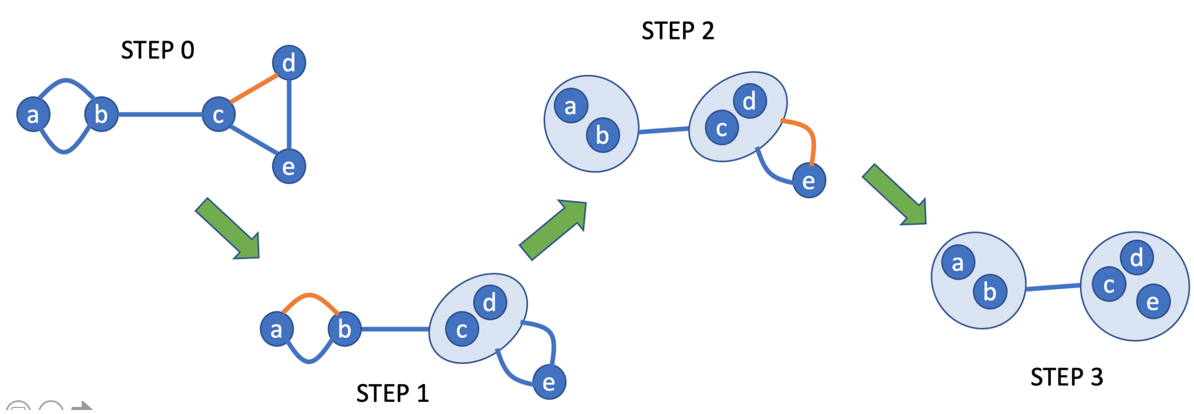
(A few slides to go over the solutions).

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

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?

Group Work: Solutions

The answer is $1/6, 1/5, 1/3$.

(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

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`

   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, click the button "1' in the "Numbers Poll", or if you get stuck, register that in the "Status Poll"

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, click the button "2' in the "Numbers Poll", or if you get stuck, register that in the "Status Poll"

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, click the button "3' in the "Numbers Poll", or if you get stuck, register that in the "Status Poll"

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, click the button "4' in the "Numbers Poll", or if you get stuck, register that in the "Status Poll"

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

   > 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?

   > Assuming that a single run of the basic Karger algorithm has success probability $O(1/n^2)$, can you show that *any* way of creating a multi-step modified Karger algorithm will still need to contract at least $\Theta(n^2)$ edges to achieve success probability at least 0.999?

Group Work: Solutions

1. The probability that the Modified Karger's algorithm fails is the probability that the first step fails, and the second step fails. By the union bound, this is bounded above by

   $$\Pr[G \to G' \text{fails}] + (\Pr[\text{ fail on } G' \ ])^k .$$

   The probability that Karger's algorithm chooses an edge crossing the min cut when reducing G to G' is at most

$$1 - \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\cdots\left(\frac{m}{m+2}\right)\left(\frac{m-1}{m+1}\right) = 1 - \frac{m(m-1)}{n(n-1)},$$

using exactly the same reasoning we used when we analyzed Karger's algorithm in the mini-lecture.

The probability that Karger's algorithm fails on G' is $1 - \frac{1}{m(m-1)}$, since this is just plugging in the result from the mini-lecture into a graph with m vertices instead of n vertices.

Adding them together, we get a bound of:

$$1 - \frac{m(m-1)}{n(n-1)} + \left(1 - \frac{2}{m(m-1)}\right)^k$$

2. Plugging in $m = \sqrt{n}$ and $k = n \log n$, and (following the hint), ignoring pesky constants, the failure probability from part 1 is at most

$$\begin{aligned} 1 - \frac{m(m+1)}{n(n-1)} + \left(1 - \frac{2}{m(m-1)}\right)^k &\approx 1 - \frac{m^2}{n^2} + (1 - 2/m^2)^k \\ &\leq 1 - \frac{n}{n^2} + \left(e^{-2/n}\right)^{n \log n} \\ &= 1 - \frac{1}{n} + e^{-2\log n} \\ &= 1 - \frac{1}{n} + \frac{1}{n^2} \\ &\leq 1 - 1/2n. \end{aligned}$$

3. If we repeat $100n$ times, the failure probability is

$$(1 - 1/(2n))^{100n} \leq e^{-50},$$

which is tiny.

4. The total number of edge contractions is $O(n^{5/2} \log n)$. This is because there are $O(n)$ contractions to reduce G to G', and $O(km) = O(n^{3/2} \log n)$ to repeat Karger k times on G'. The second part dominates, and then we repeat all of this $\Theta(n)$ times, to get $O(n^{5/2} \log n)$. This is better than $O(n^3)$!

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!