Lecture Notes CS:5360 Randomized Algorithms
Lecture 3: Aug 28, 2018
Scribe: Runxiong Dong

1 Karger's Mincut Algorithm

Algorithm 1: KARGERMINCUT(G):

$G_0 \leftarrow G$
for $i \leftarrow 1$ *to* $n-2$ **do**
  Pick an edge e_i uniformly at random from G_{i-1}
  $G_i \leftarrow CONTRACT(G_{i-1}, e_i)$
end
return The number of edges between two remaining vertices in G_{n-2}

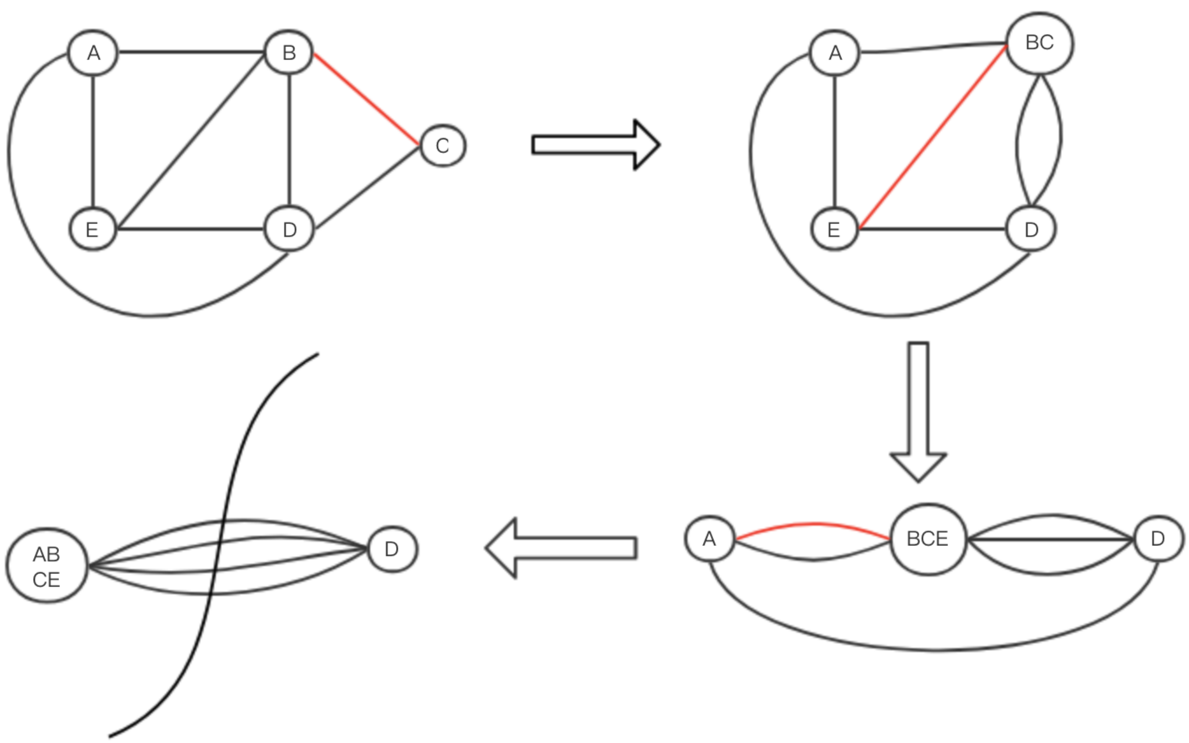

Figure 1: An example of running KARGERMINCUT() on a graph G, suppose that the red edge is selected uniformly at random in each step. In this case, the returned cut size is 4, and this is not the size of a minimum cut in G.

Analysis: Let C be the set of edges in a mincut in G. Note that G could have several mincuts, and C is one of these chosen arbitrarily. Define following events,

1. E_i = the edge e_i is not C.
2. $F_i = E_1 \cap E_2 \cap \cdots \cap E_i$. Thus, F_i is the event that none of the edges $e_1,\ e_2,\ \ldots\ e_i$ belongs to C.

Lemma 1 *Let c be the size of a mincut in G. For any edge e in G, $CONTRACT(G, e)$ has a mincut of size no less than c.*

Proof: By contradiction, suppose $G' = CONTRACT(G, e)$. Let $e = \{u, v\}$. Suppose that G' has a mincut C' of size $c' < c$. Now, we undo the $CONTRACT$ operation, i.e., separate out u and v, adding edges that were originally between u and v, and also separating out edges going to uv in G' to either u or v. Because the only new edges we added are between u and v, and they are in the same side of the cut C', the size of this cut remains unchanged. Now, the restored graph is the exactly the same as G, but it has a cut of size $c' < c$. This contracts our assumption that c is the size of mincut in G. □

Recall that F_{n-2} is the event that none of $e_1,\ e_2,\ \ldots\ e_{n-2}$ belongs to C. If F_{n-2} happens, we know the algorithm KargerMinCut() will return a right answer. Thus, we are interested in $Pr(F_{n-2})$ and in particular showing that $Pr(F_{n-2})$ is large enough.

$$\begin{aligned} Pr(F_{n-2}) &= Pr(E_1 \cap E_2 \cap \cdots \cap E_{n-2}) \\ &= Pr(E_1)Pr(E_2|E_1)Pr(E_3|E_1 \cap E_2) \cdots Pr(E_{n-2}|E_1 \cap E_2 \cap \cdots \cap E_{n-1}) \end{aligned}$$

So we now calculate $Pr(E_1)$ using the fact that $Pr(E_1) = 1 - Pr(\overline{E_1})$. Since $\overline{E_1}$ is teh complement of E_1, it is the event that e_1 belongs to C. Thus,

$$Pr(\overline{E_1}) = \frac{c}{\text{The number of edges in } G_0}.$$

Now note that the number of edges in G_0 is at least $\frac{n \cdot c}{2}$. This follows from the fact that since the size of a mincut in G_0 is c, every vertex in G_0 has degree at least c. Otherwise, we could have separated that vertex with degree less than c from the rest of the graph by deleting fewer than c edges. Thus, we would have a cut of size less than c in G_0. Thus,

$$Pr(\overline{E_1}) \ \leq \ \frac{c}{n \cdot c/2} = \frac{2}{n}$$

and therefore

$$Pr(E_1) \ \geq \ 1 - \frac{2}{n} = \frac{n-2}{n}.$$

We will next compute $Pr(E_2|E_1)$. We use $Pr(E_2|E_1) = 1 - Pr(\overline{E_2}|E_1)$. As before,

$$Pr(\overline{E_2}|E_1) = \frac{c}{\text{The number of edges in } G_1}.$$

The number of edges in G_1 is no less than $\frac{(n-1)\cdot c}{2}$, the argument is similar to the above for G_0. Thus,

$$\begin{aligned} Pr(\overline{E_2}|E_1) &\leq \frac{c}{(n-1)c/2} = \frac{2}{n-1} \\ Pr(E_2|E_1) &\geq 1 - \frac{2}{n-1}. \end{aligned}$$

Similarly,

$$\begin{aligned} Pr(E_3|E_1 \cap E_2) &\geq 1 - \frac{2}{n-2} \\ &\vdots \\ Pr(E_{n-2}|E_1 \cap E_2 \cap \cdots \cap E_{n-3}) &\geq 1 - \frac{2}{3} \end{aligned}$$

Then,

$$\begin{aligned} Pr(F_{n-2}) &\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right)\left(1 - \frac{2}{n-3}\right)\cdots\left(1 - \frac{2}{3}\right) \\ &\geq \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\left(\frac{n-4}{n-2}\right)\cdots\left(\frac{3-2}{3}\right) \\ &\geq \frac{2 \cdot 1}{n(n-1)} = \frac{2}{n(n-1)}. \end{aligned}$$

In other words, Karger's MinCut algorithm produces a correct answer with probability $\geq \frac{2}{n(n-1)}$. To amplify the correctness probability, we repeat the alogrithm t times. Then, return the smallest cut we found. The following shows the algorithm.

Algorithm 2: ImprovedKargerMinCut(G):

```
k ← ∞
for i ← 1 to t do
    z ← KargerMinCut(G)
    if z < k then
        k ← z
    end
end
return k
```

Analysis: Let $c =$ the size of minimal cut in graph G, k is the number return by the algorithm. Then,

$$Pr(k \neq c) \leq \left(1 - \frac{2}{n(n-1)}\right)^t$$

Here is an inequality that turns out to be super-useful:, $1 + x \leq e^x$ for all reals x. Thus, we get

$$Pr(k \neq c) \leq e^{-\frac{2t}{n(n-1)}}$$

- If we pick $t = n(n-1)$, then $Pr(k \neq c) \leq e^{-2} = \frac{1}{e^2}$.
- If we pick $t = n(n-1)\ln n$, then $Pr(k \neq c) \leq e^{-2\ln n} = \frac{1}{n^2}$.

2 Random Variables

Definition: A random variable is a variable that takes on different values with associated probabilities.

EXAMPLE:

Algorithm 3:

while *not* $\frac{|L|}{3} \leq |L_1| \leq \frac{2|L|}{3}$ **do**
  $(L_1, L_2) \leftarrow$ RANDOMIZEDPARTITION(L)
end
return (L_1, L_2)

Let I = number of iterations of the Algorihtm 3. The following is the probability distribution of I.

i	1	2	3	$\cdots$	i	$\cdots$
$Pr(I = i)$	$\frac{1}{3}$	$\left(\frac{2}{3}\right)\frac{1}{3}$	$\left(\frac{2}{3}\right)^2\frac{1}{3}$	$\cdots$	$\left(\frac{2}{3}\right)^{i-1}\frac{1}{3}$	$\cdots$

Table 1: The probability distribution of I

This distribution is a geometric distribution with parameter $p = \frac{1}{3}$.

Definition: The expectation of a discrete random variable X denoted $E[X]$ is

$$E[X] = \sum_i i \cdot Pr(X = i).$$

Following the definition, we can compute $E[I]$ by the following formula,

$$E[I] = \sum_{i \geq 1} i \left(\frac{2}{3}\right)^i \frac{1}{3} = \frac{1}{3} \sum_{i \geq 1} i \left(\frac{2}{3}\right)^i.$$

$$S = 1 \cdot (\frac{2}{3})^0 + 2 \cdot (\frac{2}{3})^1 + 3 \cdot (\frac{2}{3})^2 + \cdots + i \cdot (\frac{2}{3})^{i-1} + \cdots \quad (1)$$

$$\frac{2}{3} S = 1 \cdot (\frac{2}{3})^1 + 2 \cdot (\frac{2}{3})^2 + \cdots + (i-1) \cdot (\frac{2}{3})^{i-1} + i \cdot (\frac{2}{3})^i + \cdots \quad (2)$$

Subtracting (2) from (1), we get

$$\frac{1}{3} S = \left(\frac{2}{3}\right)^0 + \left(\frac{2}{3}\right)^1 + \left(\frac{2}{3}\right)^2 + \cdots = \frac{1}{1 - 2/3} = 3.$$

Thus, $S = 9$ and $E[I] = 3$. In general, $E[X] = 1/p$, for a geometric random variable with parameter p.

3 Geometric Random Variable

Definition: A geometric random variable X with parameter p, $0 < p < 1$, has the following probability distribution over $i = 1,\ 2,\ 3,\ \cdots$

$$Pr(X = i) = (1-p)^{i-1}p.$$

It is easy to show the the probabilities indeed add upto 1. In other words,

$$\sum_{i\geq 1} Pr(X = i) = 1.$$

$$\begin{aligned}
\sum_{i\geq 1} Pr(X = i) &= p + p(1-p) + p(1-p)^2 + \cdots \\
&= p\left(1 + (1-p) + (1-p)^2 + \cdots\right) \\
&= p(\frac{1}{1-(1-p)}) \\
&= p \cdot \frac{1}{p} \\
&= 1.
\end{aligned}$$

From line 2 to line 3, it is because the fact that $\frac{1}{1-x} = 1 + x^1 + x^2 + \cdots$, for $0 < x < 1$.

Lemma 2 *Let X be a discrete random variable that takes on values $i = 1,\ 2,\ 3,\ \cdots$.*

$$E[X] = \sum_{i\geq 1} Pr(X \geq i)$$

EXAMPLE: let X be a geometric random variable with parameter p. $Pr(X \geq t) = (1-p)^{t-1}$, since $Pr(X \geq t)$ means that the first $t-1$ trails fail. Then, using this fact and Lemma 2, we can compute $E[X]$ as following,

$$\begin{aligned}
E[X] &= \sum_{i\geq 1} Pr(X \geq i) \\
&= \sum_{i\geq 1} (1-p)^{i-1} \\
&= \frac{1}{1-(1-p)} \\
&= \frac{1}{p}.
\end{aligned}$$

4 Linearity of Expectation

Theorem 3 *Let $X_1,\ X_2,\ X_3,\ X_4,\ \cdots, X_n$ be random variables with finite expectations. Then,*

$$E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i]$$

Important note: no independence needed!

Definition: Independence of random variables, random variables X and Y are independent if all values x, y,

$$Pr(X = x \cap Y = y) = Pr(X = x)Pr(Y = y).$$

EXAMPLE: let X = sum of two 6-sided dice outcomes. To compute $E[X]$, we have two ways, first directly by definition. Looking at the following probability distribution table. So we can

i	2	3	4	$\cdots$	11	12
$Pr(X = i)$	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\cdots$	$\frac{2}{36}$	$\frac{1}{36}$

Table 2: The probability distribution of X

calculate $E[X]$ as follows.

$$E[X] = \sum_{i=2}^{12} Pr(X = i) \cdot i$$

However, this sum is kind of tedious to deal with. So let us look at this another way, using linearity of expectation. For $i = 1,\ 2$: X_i = outcome of i^{th} dice. Then,

$$\begin{aligned}
X &= X_1 + X_2 \\
E[X] &= E[X_1 + X_2] \\
&= E[X_1] + E[X_2] \\
&= 2\ E[X_1] \\
&= 2 \cdot \frac{1}{6} \cdot (1 + 2 + 3 + 4 + 5 + 6) \\
&= 7
\end{aligned}$$

Theorem 4 *Let X, Y be random variables with finite expectation, then $E[X + Y] = E[X] + E[Y]$.*

Proof:

$$
\begin{aligned}
E[X+Y] &= \sum_i \sum_j (i+j) Pr(X=i \cap Y=j) \\
&= \sum_i \sum_j i \cdot Pr(X=i \cap Y=j) + \sum_i \sum_j j \cdot Pr(X=i \cap Y=j) \\
&= \sum_i i \sum_j Pr(X=i \cap Y=j) + \sum_j j \sum_i Pr(X=i \cap Y=j) \\
&= \sum_i i \cdot Pr(X=i) + \sum_j j \cdot Pr(Y=j) \\
&= E[X] + E[Y]
\end{aligned}
$$

□

5 Coupon Collector's Problem

Suppose each box of cereal contain one of n distinct coupons, and assume a coupon in a box is chosen uniformly at random from n distinct coupons. You win once you obtain at least one of every distinct type of coupon. The question is how many boxes of cereal you must buy to win. Let X = the number of boxes to buy for winning. Apparently, X is a random variable, you can be luck enough so that only buy n boxes to win, but this is just high unlikely when n is large. We are interested in $E[X]$. Let X_i = the number of boxes you need to buy while you have $i-1$ distinct coupons.

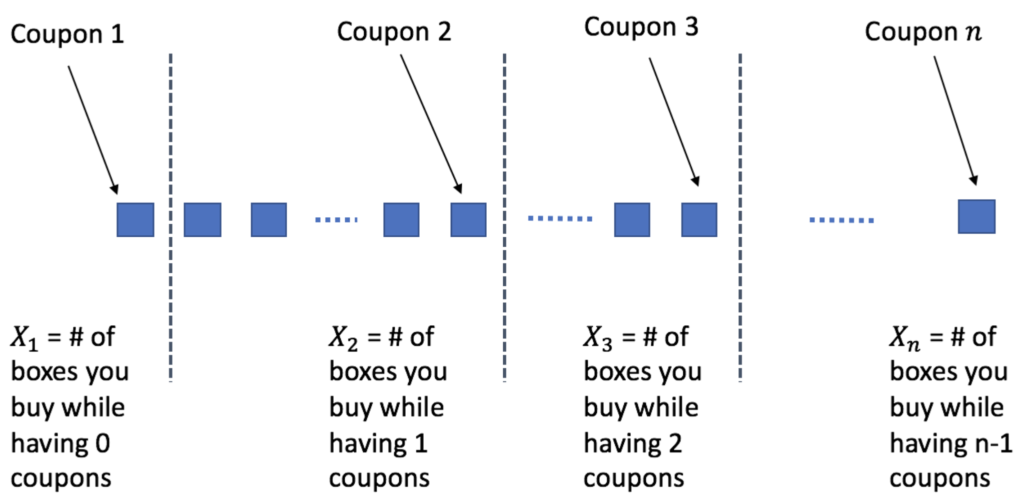

Figure 2: The visualization of buying enough cereal boxes to win

By the definition of X_i, $i = 1,\ 2,\ \cdots, n$.

$$X = X_1 + X_2 + \cdots + X_n$$

X_i is a geometric random variable with $p = \frac{n-(i-1)}{n}$. Thinking of X_i, when you have $i-1$ distinct coupons, the probability to obtain a new distinct coupon is $1 - \frac{i-1}{n} = \frac{n-(i-1)}{n}$, every time you buy one cereal box. Therefore,

$$E[X_i] = \frac{1}{p} = \frac{n}{n-(i-1)}$$

$$\begin{aligned} E[X] &= E[X_1 + X_2 + X_3 + \cdots + X_n] \\ &= E[X_1] + E[X_2] + E[X_3] + \cdots + E[X_n] \\ &= \sum_{i=1}^{n} \frac{n}{n-i+1} \\ &= n \sum_{i=1}^{n} \frac{1}{n-i+1} \\ &= n \ (1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}) \end{aligned}$$

H_n is the n^{th} harmonic number,

$$\begin{aligned} H_n &= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \\ &= \ln n + O(1) \end{aligned}$$

6 Randomized QuickSort

Algorithm 4: QuickSort($L[1..n]$):

```
/* Assume L contains distinct elements, which can be removed later
c ← some constant
if n ≤ c then
    BubbleSort(L)
    return L
else
    p ← index chosen from {1, 2, ,··· ,n} uniformly at random
    L1, L2 ← ∅
    for i ← 1 to n do
        if L[i] < L[p] then
            L1 ← L1 ∘ L[i]
        end
        if L[i] > L[p] then
            L2 ← L2 ∘ L[i]
        end
    end
    QuickSort(L1)
    QuickSort(L2)
end
return L1 ∘ L[p] ∘ L2
```

In the worst case, QuickSort form $\Omega(n^2)$ comparison. If the pivot is always the median of current array, then

$$T(n) = 2\ T(\lceil \frac{n}{2} \rceil) + O(n)$$
$$T(n) = O(n \log n)$$

This is the best case, and we don't have to be that lucky, if pivot partition L into sublists L_1, L_2 such that

$$\frac{|L|}{3} \leq |L_1| \leq \frac{2|L|}{3}$$

we can also have a good running time on expectation. If the above relation fulfilled, then

$$T(n) < T(\frac{n}{3}) + T(\frac{2n}{3}) + O(n)$$
$$\Rightarrow \quad T(n) = O(n \log n)$$

$L_1, L_2 \leftarrow$RandomizedPartition(L) is a way to achieve it.