CS240 Homework 4

Jiang Caiwen Student ID: 2019233157

June 17, 2020

Problem 1

As before, all G are multigraphs.
Fix a min-cut C in the original multigraph G. By the same analysis as in the case of f, we have

$$\begin{aligned}
&Pr[\text{C survives all contractions in f(G,t)}] \\
&= \prod_{i=1}^{n-t} Pr[\text{C survives the i-th contraction}|\text{C survives the first (i-1)-th contraction}] \\
&\geq \prod_{i=1}^{n-t} (1 - \frac{2}{n-i+1}) \\
&= \prod_{k=t+1}^{n} \frac{k-2}{k} \\
&= \frac{t(t-1)}{n(n-1)}
\end{aligned}$$

When $t = [1 + n/\sqrt{2}]$, this probability is at least 1/2. The choice of t is due to our purpose to make this probability at least 1/2. You will see this is crucial in the following analysis of accuracy.
We denote by A and B the following eventd:
A: C survives all contractions in f(G,t);
B : size of min-cut is unchanged after f(G,t);
Clearly, A implies B and by above analysis $Pr[B] \geq Pr[A] \geq \frac{1}{2}$
We denote by $p(n)$ the lower bound on the probability that randomized algorithm succeeds for a multigraph of n vertices, that is

$$p(n) = \min_{G:|V|=n} Pr[\text{randomized algorithm returns a min-cut in G}]$$

Suppose that G is the multigraph that achieves the minimum in above definition. The following recurrence holds for $p(n)$:

$$\begin{aligned}
p(n) &= Pr[\text{randomized algorithm returns a min-cut in G}] \\
&= Pr[\text{a min-cut of G is returned by } cut2(G_1) \text{ or } cut2(G_2)] \\
&\geq 1 - (1 - Pr[B \cup cut2(G_1) \text{ return a min-cut in } G_1])^2 \\
&\geq 1 - (1 - Pr[A \cup cut2(G_1) \text{ return a min-cut in } G_1])^2 \\
&= 1 - (1 - Pr[A]Pr[cut2(G_1) \text{ return a min-cut in } G_1|A])^2 \\
&\geq 1 - (1 - \frac{1}{2}p([1 + n/\sqrt{2}]))^2
\end{aligned}$$

where A and B are defined as above such that $Pr[A] \geq 1/2$
The base case is that $p(n) = 1$ for $n \leq 6$. By induction it is easy to prove that

$$p(n) = \Omega(\frac{1}{logn})$$

Problem 2

1. Let G be a graph, and let $C_1, ...C_r$ denote all its global min-cut, let E_i denote the event that C_i is returned by the Contraction Algorithm, let $E = \sum_{i=1}^{r} E_i$denote the event that the algorithm returns any global min-cut.
We know the contraction algorithm returns a min-cut with $prob \geq 2/n^2$, so

$$Pr[E] \geq 2/n(n-1)$$
$$Pr[E_i] \geq 2/n(n-1)$$

Then we have

$$Pr[E] = Pr[\sum_{i=1}^{r} E_i] = \sum_{i=1}^{r} Pr[E_i] \geq r\frac{2}{n(n-1)}$$

But clearly $Pr[E] \leq 1$, and so we must have $r \leq \frac{n(n-1)}{2}$.

2. The probability that the algorithm finds the minimum cut in G is at least $2/n^2$.

Algorithm 1 Karger's Randomized Global Min-Cut

```
repeat
    Choose an edge {u, v} uniformly at random from E.
    Contract the vertices u and v to a super-vertex w.
    Keep parallel edges but remove self-loops.
until G has only 2 vertices.
Report the corresponding cut.
```

Now we have to prove the correctness of the algorithm:
The probability of finding a min-cut seems low at face value, and goes to zero as $n \to \infty$. However, if we run the algorithm t times and output the smallest min-cut found over all runs, the probability of success will be at least

$$1 - (1 - \frac{2}{n^2})^t$$

So if we set $t = cn^2$ for some constant c, the probability of failure will be at most $(1 - \frac{2}{n^2})^{cn^2} \leq e^{-2c}$. Thus to ensure the probability of failure is smaller than some fixed constant ϵ, it is only necessary to run the algorithm $\frac{1}{2}log(\frac{1}{\epsilon})n^2$ times.

Problem 3

Flip a coin, and set each variable true with probability 1/2, independently for each variable.

Appro-Max3SAT

---

for i = 1 to n

do Flip a fair coin

if Heads

then $x_i \leftarrow 1$
esle $x_i \leftarrow 0$

return x

The running time of Approx-Max3SAT is O(n)
Approx-Max3SAT is 7/8-approximate.
Proof:
Define the random variable:

$$Z_j = \begin{cases} 1, \text{if clause } C_j \text{ is satisfied} \\ 0, otherwise \end{cases} \tag{1}$$

Then, $Z = \sum_{j=1}^{k} Z_j$ is the number of clauses satisfied.
The expected number of clauses satisfied is

$$E[Z] = \sum_{j=1}^{k} E[Z_j] = \sum_{j=1}^{k} Pr(C_j \text{is satisfied}) = \frac{7}{8}k$$

because a random variable is at least its expectation some of the time, so we can get
For any instance of 3-SAT, there exists a truth assignment that satisfies at least 7/8 of the clauses.

Problem 4

The number of dice is represented by the H and T combinations of coins

$$HTH \rightarrow 1$$

$$HTT \rightarrow 2$$

$$THH \rightarrow 3$$

$$THT \rightarrow 4$$

$$TTH \rightarrow 5$$

$$TTT \rightarrow 6$$

$$HHH \rightarrow Null$$

$$HHT \rightarrow Null$$

We have a 3/4 chance of being able to get one of the 6 good results (HTH, HTT, THH, THT, TTH, TTT) that yields a random number from 1 to 6, and a 1/4 chance of having to reflip the coins (HHH or HHT).

The key is that for those two results to be discarded, we only needed to flip 2 coins to determine that!
Let X is the number of coins required

$$E[X] = 3/4 \times 3 + 1/4 \times (2 + E[X])$$

So, $E[X] = 11/3$

Problem 5

For every array element and for every recursion level t, let X, t be the indicator random variable for the event of choosing a bad pivot in s subarray at level t:

$$X_{\alpha,t} = \begin{cases} 1, \text{iif, at level t, quicksort chose a bad pivot for the subarray containing } \alpha \\ 0, otherwise \end{cases} \tag{2}$$

fix α and take $T = clogn$ for constant c, δ to be chosen. We apply a Chernoff bound and simplify the exponential:

$$\begin{aligned} & Pr[\sum_{i=1}^{T} X_{\alpha,i} \geq (1+\delta)E[\sum_{i=1}^{T} X_{\alpha,i}]] \\ & \leq e^{-\delta^2 E[\sum_{i=1}^{T} X_{\alpha,i}]/2} \\ & = e^{-\delta^2 \sum_{i=1}^{T} E[X_{\alpha,i}]/2} \\ & \leq e^{-\delta^2 T/2} \\ & = n^{-\delta^2 c} \end{aligned}$$

Taking $\delta = 1$, we have

$$Pr[\sum_{i=1}^{T} X_{\alpha,i} \geq 2E[\sum_{i=1}^{T} X_{\alpha,i}]] \leq \frac{1}{n^c}$$

Thus we can get
$Pr[\text{Runtime of quicksort} < nlongn] \leq 1 - \frac{1}{n^c}$