Randomized Algorithms
Assignment 1

Nikolaj Dybdahl Rathcke (rfq695)
Victor Petren Bach Hansen (grnXXX)

April 27, 2016

1 Summary of covered material

TODO

2 Exercise 1.2

To prove there are inputs where the possibility of finding a min-cut is exponentially low, we just need to find some graph family $\mathbb{F}$ which has an exponentially low probability.
Consider the graphs which consist of two complete graphs of size n, K_n, which are connected by a single edge. For analysis sake, both complete graphs must have size n. Figure 1 shows the graph type with $n = 3$.

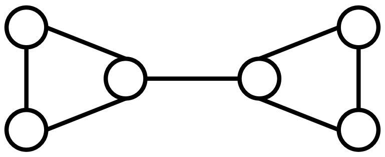

Figure 1: Two complete graphs of size n connected by a single edge

Now we want to show that the probability of getting a correct result with this type of graph is exponentially small when we contract vertices instead of edges. We see that if contract two vertices that belong to different K_n, it will join the graphs in a one vertex. This means our optimal cut no longer has size 1. To see that the probability of this *not* happening in any of the $n-2$ iteration, call the event of successfully contracting each complete graph into a single vertex (that the remaining two nodes are the contractions of all the nodes from their original K_n) for A. We also know that for a result to be correct, each of the remaining nodes are the contraction of exactly n nodes (not nessecarily a correct solution). Call this type of event for B. Now, the number of events of type B are actually the number of ways we can pick n nodes out of $2n$ nodes, namely $\binom{2n}{n}$. Of this number, only 2 of them are events of type A. These are the cases where we pick the n nodes from either the first or the second complete graph. So we have the probability of success (event A) is:

$$\begin{aligned}
\mathbb{P}[A] &= \mathbb{P}[A|B]\,[B] \\
&\leq \frac{2}{\binom{2n}{n}} && \text{(Bounding the probability by removing an event } B) \\
&\leq \frac{2}{\left(\frac{2ne}{n}\right)^n} && \left(\text{As } \tbinom{n}{k} \leq \left(\tfrac{ne}{k}\right)^k.\right) \\
&= \frac{2}{(2e)^n}
\end{aligned}$$

Which shows the probability is exponentially small. Note we remove the probability of event B happening in the first place. This is because by removing it, it only makes the probability greater - an upper bound. It also allows us to disregard the probability of event B happening in the first place to avoid counting all possible outcomes.

3 Exercise 1.3

To get an expected running time of $\frac{T(n)+t(n)}{\gamma(N)}$, we do the obvious. We run the Monte Carlo algorithm in $T(n)$ time and then validate it in $t(n)$ time. The fraction comes from getting the probability of getting the result right. We can expect that we need to run the Monte Carlo algorithm $1/\gamma(n)$ times and therefore we can expected the validation algorithm $1/\gamma(n)$ times. Multiplying the running times for two algorithm into the numerator yields the desired expected running time.

4 Problem 1.1

4.1 (a)

Let T be tails and H be heads. We devise a scheme that tosses the coin twice. If the coin has the same result, then we disregard the result and toss it twice again. If the results are different, we pick the first (or second, as long as it is chosen beforehand) result. The reason this generates unbiased coin-flips is that the probability of gettings TH is the same as getting HT since they are independent flips. The other two scenarios are disregarded, meaning the probability the choice comes from either of these events is 0. Thus we have a 50 percent probability on the remaining events. Note that $p > 0$ or we will end up throwing the coin forever.
If the probability of getting H is p, then obviously the probability of T is $1 - p$. Now, the expected number of rounds (throwing it twice) before getting a result must be $\frac{1}{2p(1-p)}$ - the probability that either TH or HT happens . If we want to count the number of throws of the coin, we simply multiply the result by 2, yielding:

$$\mathbb{E}\left[\#\text{ of throws}\right] = \frac{2}{2p(1-p)} = \frac{1}{p(1-p)}$$

which is what we wanted to show.

4.2 (b)

TODO

5 Problem 1.4

5.1 (a)

TODO

5.2 (b)

TODO

5.3 (c)

TODO

6 Problem 1.8

6.1 (a)

TODO

6.2 (b)

TODO