CS 6150: Homework 2

Christopher Mertin

(Worked with Johnny Walker, Jack Daniels, and Jim Bean)

Due Date: October 1, 2015

This assignment has 7 questions, for a total of 100 points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

Question	Points	Score
Tail Bounds I	20	
Tail Bounds II	15	
Medians	10	
Amplification	10	
The "median" trick	15	
Hashing with open addressing	15	
Derandomizing MAX CUT	15	
Total:	100	

Equation Definitions

Expectation Value:

$$\langle X \rangle = \sum_{i=0}^{\infty} xP\left(x\right)$$

Variance:

$$\sigma^2\left(X\right) = \left\langle X^2 \right\rangle - \left\langle X \right\rangle^2$$

Standard Deviation:

$$\sigma(X) = \sqrt{\sigma^2\left(X\right)}$$

Here are the three increasingly-strong inequalities we use to analyze the tail of a distribution.

Lemma 0.1 (Markov)**.** *Let X be a random variable taking nonnegative values. Then for any $a > 0$,*

$$\Pr(X \geq a) \leq \frac{\langle X \rangle}{a}$$

Lemma 0.2 (Chebyshev)**.** *Let X be a random variable with bounded variance. Then*

$$\Pr(|X - \langle X \rangle| \geq a) \leq \frac{\sigma^2(X)}{a^2}$$

Lemma 0.3 (Chernoff)**.** *Let $X_1, \ldots, X_n$ be independent random variables taking the values $0, 1$ with $E[X_i] = \Pr(X_i = 1) = p_i$. Let $\mu = \sum_i p_i$. Then for any $\delta > 0$,*

$$\Pr(X \geq \mu(1+\delta)) \leq \exp\left(-\frac{\mu\delta^2}{3}\right)$$

$$\Pr(X \leq \mu(1-\delta)) \leq \exp\left(-\frac{\mu\delta^2}{2}\right)$$

Question 1: Tail Bounds I .... **[20]**

(a) [**5**] The Markov inequality yields an *upper* bound on the probability of going far away from the mean. It is reasonable to ask whether we can do better. Show that you can't, by constructing some nonnegative random variable X and a value $a > 0$ such that

$$\Pr(X \geq a) = \frac{\langle X \rangle}{a}$$

Solution:

The definition of the expected value is as follows:

$$\langle X \rangle = \sum_{i=0}^{\infty} XP(X) = \sum_{i=0}^{\infty} P(X \geq i) \tag{1}$$

Since the range of the set is finite and of size a, and since $a > 0$

$$\langle X \rangle \geq \sum_{i=0}^{a-1} P(X \geq i) \tag{2}$$

And since $i < a$, we can use that in the probability

$$\langle X \rangle = a \cdot P(X \geq a) \tag{3}$$

$$\frac{\langle X \rangle}{a} = P(X \geq a) \tag{4}$$

(b) [**5**] Suppose we roll a fair die 100 times. Determine the probability that the sum of the rolls is at least 400 (using Markov's inequality) and the probability it is not in the range $[301, 399]$ (using Chebyshev's inequality).

Solution:

Expectation value of a dice roll is defined as

$$\langle X \rangle = \sum_{i=0}^{\infty} XP\left(X\right) = \sum_{i=1}^{6} i \cdot P(i) = \frac{1}{6}(1 + 2 + \cdots + 6) = \frac{21}{6} = 3.5 \tag{5}$$

That means that the expected value of 100 rolls is

$$\langle X \rangle_{100} = 350 \tag{6}$$

Using the Markov Method to calculate the probability of the sum being more than 400

$$P(X \geq 400) = \frac{350}{400} \tag{7}$$

Now for in the probability **NOT** in the range of $[301, 399]$. The first the squared expectation value is needed for a single dice roll

$$\left\langle X^2 \right\rangle = \frac{1}{6} \sum_{i=1}^{6} i^2 = \frac{91}{6} \tag{8}$$

Where the variance can also be calculated from this

$$\sigma^2\left(X\right) = \frac{91}{6} - 3.5^2 = 2.917 \tag{9}$$

$$\sigma^2\left(X\right)_{100} = 291.7 \tag{10}$$

And the use of the Chebyshev Inequality leads to using $a = 50$ since the range that we are wanting is 100 $[301, 399]$ which is 50 in each direction (since the probability is denoted in each direction).

$$P\left(|X - \langle X \rangle| \geq 50\right) = \frac{291.7}{50^2} \approx 0.1167 \tag{11}$$

(c) [**5**] Suppose we're given an algorithm that takes as input a string of n bits. We are told that the expected running time is $O(n^2)$ if the bits are chosen independently and uniformly at random. What can we say about the *worst-case* behavior of the algorithm on inputs of size n (using Markov's inequality).

Solution:

$$P\left(X \geq \log(n)\right) \leq \frac{\mathcal{O}\left(n^2\right)}{\log(n)} \tag{12}$$

(d) **[5]** We prove Chebyshev's inequality by applying Markov's inequality to the positive random variable $Y = (X - \langle X \rangle)^2$. Can you generalize Chebyshev's inequality to higher moments of X (i.e values $E[X^k]$ for large k). In particular, set $k > 2$ to be some **even** number and derive a Chebyshev-like bound for the tail. What do you notice about such a bound?

Solution:

From the definition of Y in the question, the expected value of it is

$$\langle Y \rangle = \left\langle |X - \langle X \rangle|^k \right\rangle \tag{13}$$

From the Markov Inequality

$$P\left(Y \geq a^k\right) = \frac{\left\langle |X - \langle X \rangle|^k \right\rangle}{a^k} \tag{14}$$

$$P\left(|X - \langle X \rangle|^k \geq a\right) = \frac{\left\langle |X - \langle X \rangle|^k \right\rangle}{a^k} \tag{15}$$

Which is the same thing as the following relation by the definition of the Chebyshev Inequality

$$P\left(|X - \langle X \rangle|^k \geq a \left\langle |X - \langle X \rangle|^k \right\rangle\right) = \frac{1}{a^k} \tag{16}$$

As seen in Equation (16), as $k \to \infty$, the probability of the bound goes to zero

Question 2: Tail Bounds II .... **[15]**

(a) **[5]** Repeat the Chebyshev analysis for the fair die above but using a Chernoff bound instead. Since your random variables are now not $0 - 1$, you will need a slightly different version of the Chernoff bound called a Hoeffding bound:

Lemma 0.4 (Hoeffding)**.** *Let $X_1, \ldots, X_n$ be independent random variables where $a_i \leq X_i \leq b_i$. Let $S = \sum X_i$. Then*

$$\Pr\left(|S - \langle S \rangle\,| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_i (b_i - a_i)^2}\right)$$

Solution:

$$P\left(|S - \langle S \rangle| \geq 50\right) = 2e^{\alpha} \tag{17}$$

$$\alpha = -\frac{-2 \cdot 50^2}{100 \cdot 5^2} = -2 \tag{18}$$

$$P\left(|S - \langle S \rangle| \geq 50\right) = 2e^{-2} \approx 0.2707 \tag{19}$$

(b) **[10]** Suppose you toss $2n$ fair coins and compute the difference between the number of heads and tails. On average, this difference is zero. But what is the probability that the difference is more than t? (**Hint:** Consider just the number of heads).

Solution:

From the Chernoff Inequality

$$P\left(X \geq \mu(1+\delta)\right) \leq \exp\left(-\frac{\mu\delta^2}{3}\right) \tag{20}$$

The expectation value can be calculated for μ, which can be done by just assuming the number of heads

$$\langle X \rangle = P\left(X_i = 1\right) = \frac{2n}{2} = n \tag{21}$$

Set $\delta = t$ and $\mu = n$

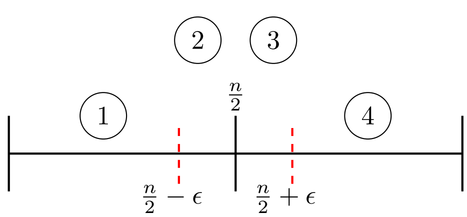
$$P\left(X \geq n(1+t)\right) \leq \exp\left(\frac{-nt^2}{3}\right) \tag{22}$$

Question 3: Medians . . . . **[10]**

We saw in class that taking a median of three samples suffices to obtain a good approximation to the rank of the median. Assume that our goal is to return an element whose with rank r such that $|r - n/2| \leq \epsilon n/2$. Assume also that we desire this result with probability $1 - \delta$.

We can take a sample of size t and select the median as our result. What value of t must we pick (as a function of ϵ, δ and n)? Do you notice something interesting/unusual?

Solution:



The distribution of numbers can be broken up into 4 segments. To get the median by the rank, it needs to be such that most of the numbers that are chosen are in sections 2 or 3. However, there is a problem that occurs when more than $\frac{t}{2}$ terms are in 1 or 4. As stated in the problem, we want

$$\left|r - \frac{n}{2}\right| \leq \epsilon\frac{n}{2} \tag{23}$$

$$\frac{n}{2} - \epsilon\frac{n}{2} \leq r \leq \frac{n}{2} + \epsilon\frac{n}{2} \tag{24}$$

Which comes about from removing the absolute value sign. The probability that the element is in reagion 1 is the following

$$P(\text{rank}(i) \in 1) = \frac{\frac{n}{2} + \epsilon\frac{n}{2}}{n} = \left(\frac{1}{2} - \frac{\epsilon}{2}\right) \tag{25}$$

This is the probability for one element to be found in region 1. What is needed is for more than $\frac{t}{2}$ to be in region 1

$$\xi = \text{rank}(i) \in 1$$

$$P\left(\xi \geq \frac{t}{2}\right) = \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{\frac{t}{2}} + \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{\frac{t}{2}+1} + \cdots + \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{t} \tag{26}$$

$$P\left(\xi \geq \frac{t}{2}\right) = \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{t} \left[\sum_{i=0}^{\frac{t}{2}} \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{i}\right] \tag{27}$$

$$= \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{\frac{t}{2}} \left[\frac{1 - \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{\frac{t}{2}+1}}{1 - \left(\frac{1}{2} - \frac{\epsilon}{2}\right)}\right] \tag{28}$$

Which is the probability of more than $\frac{t}{2}$ elements being in region 1. δ can be calculated from this. By revisiting the definition of probability in the question

$$P\left(\left|r - \frac{n}{2}\right| < \epsilon\frac{n}{2}\right) \leq 1 - \delta \tag{29}$$

$$\delta = 1 - P\left(\left|r - \frac{n}{2}\right| < \epsilon\frac{n}{2}\right) \tag{30}$$

$$\delta = P\left(\left|r - \frac{n}{2}\right| \geq \epsilon\frac{n}{2}\right) \tag{31}$$

But what is wanted is the value in regions 1 and 4, and since they're symmetric it can just be multiplied by 2, giving

$$\delta = 2P\left(\left|r - \frac{n}{2}\right| \geq \epsilon\frac{n}{2}\right) \tag{32}$$

Using relation in Equation (28) and plugging it in, we get

$$\delta = 2 \cdot \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{\frac{t}{2}} \left[\frac{1 - \left(\frac{1}{2} - \frac{\epsilon}{2}\right)^{\frac{t}{2}+1}}{1 - \left(\frac{1}{2} - \frac{\epsilon}{2}\right)}\right] \tag{33}$$

Now, for solving for t

$$\xi = \left(\frac{1}{2} - \frac{\epsilon}{2}\right) \tag{34}$$

$$\delta = 2\xi^{\frac{t}{2}} \left[\frac{1 - \xi^{\frac{t}{2}+1}}{1 - \xi}\right] \tag{35}$$

$$\underbrace{\frac{\delta}{2}(1 - \xi)}_{\Omega} = \xi^{\frac{t}{2}} - \xi^{t+1} \tag{36}$$

Using Mathematica to solve this

$$t = \frac{2\log\left[\frac{\xi\sqrt{\frac{4\Omega\xi+1}{\xi^2}}-1}{2\xi}\right]}{\log(\xi)} \tag{37}$$

Question 4: Amplification .... **[10]**

Consider a decision problem f (i.e one where the output is either zero or one). Suppose we are given a randomized algorithm A that on input x has the following properties:

- If $f(x) = 0$, then $A(x) = 0$
- If $f(x) = 1$, then $A(x) = 1$ with probability at least $2/3$.

Such an algorithm is said to have *one-sided error*, and as we've discussed in class we can amplify the probability of being correct merely by repeating it a number of times, and returning 1 if we ever see a 1, returning 0 otherwise.

But suppose our algorithm instead had the following properties:

- If $f(x) = 0$, then $A(x) = 0$ with probability at least $2/3$
- If $f(x) = 1$, then $A(x) = 1$ with probability at least $2/3$.

In other words, it has *two-sided* error rather than one-sided error.

(a) **[5]** What is the probability of success (in either case) if we implement the above procedure (returning a 1 if we ever see a 1, else returning a zero) after k iterations?

Solution:

The probability of success (S) is defined as

$$\begin{aligned} P(Success) &= P(S \cap f = 0) + P(S \cap f = 1) \qquad (38) \\ &= P(S|f = 0) \cdot \underbrace{P(f = 0)}_{Assume\ \frac{1}{2}} + P(S|f = 1) \cdot \underbrace{P(f = 1)}_{Assume\ \frac{1}{2}} \qquad (39) \end{aligned}$$

The probability of getting all 0 for $f = 0$ is $\left(\frac{2}{3}\right)^k$, while the probability of getting all 1 for $f = 1$ is 1 minus the probability of getting all zero's for $f = 1$, ie $\left(1 - \frac{1}{3^k}\right)$

$$= \left(\frac{2}{3}\right)^k \cdot \frac{1}{2} + \frac{1}{2} \cdot \left(1 - \frac{1}{3^k}\right) \tag{40}$$

(b) [**5**] What is the probability of success (in either case) of the following procedure: Run the algorithm k times and return the majority answer.

Solution:

$$P\left(X_0 > \frac{k}{2}\right) \geq \sum_{i=1}^{\frac{k}{2}} \left(\frac{2}{3}\right)^{\frac{k}{2}+i} \left(\frac{1}{3}\right)^{\frac{k}{2}-i} \tag{41}$$

$$\geq \sum_{i=1}^{\frac{k}{2}} \frac{2^{\frac{k}{2}+i}}{3^k} \tag{42}$$

$$\geq \frac{1}{3^k} \sum_{i=1}^{\frac{k}{2}} 2^{\frac{k}{2}+i} = \frac{1}{3^k}\left[2^{\frac{k}{2}+1}\right]\left[2^{\frac{k}{2}}-1\right] \tag{43}$$

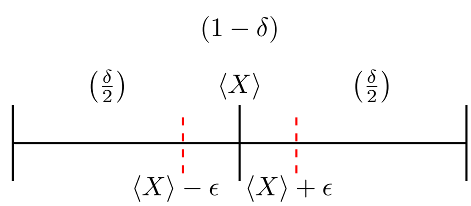
Question 5: The "median" trick . . . . [**15**]

Suppose we have an algorithm that generates independent samples $X_1, \ldots$ of a random variable X. A natural estimate of the value of X is the mean $S = \sum_{i=1}^{t} X_i/t$ for some value of t. What is the probability that S is close to $\langle X \rangle$? Specifically, what is the probability that $|S - \langle X \rangle| \leq \epsilon \langle X \rangle$?

We can analyze this if we know that $\sigma^2(X)$ is bounded. In particular, set $r = \sqrt{\sigma^2(X)}/\langle X \rangle$.

(a) [**5**] Show that if we set $t = O(r^2/\epsilon^2\delta)$ then the probability is at least $1-\delta$ (or in other words the probability of the difference *exceeding* $\epsilon \langle X \rangle$ is at most δ).

Solution: The figure below shows how the range of values can be partitioned. The values in parenthesis over each region is the probability of being in that region, which is what we're trying to prove.



For simplicity, set the range to be the following variable

$$\alpha = \langle X \rangle - \epsilon \langle X \rangle < \langle X \rangle < \langle X \rangle + \epsilon \langle X \rangle \tag{44}$$

For the median to be outside α, at least one has to be outside α. The probability of choosing one outside of α is the following. From the defintion we have

$$P\left(|X - \langle X \rangle| \geq a\right) \leq \frac{\sigma^2(X)}{a^2} \tag{45}$$

Plugging in the relevant info

$$P\left(|X - \langle X \rangle| \geq \epsilon \langle X \rangle\right) \geq \frac{\sigma^2(X)}{\epsilon^2 \langle X \rangle^2} \tag{46}$$

Which is the probability that it is not in the range. However, we have a set t, so the probability of one value of t not being in the range is, and by plugging in $t = \mathcal{O}\left(\frac{r^2}{\epsilon^2 \delta}\right)$

$$P\left(\left|X - \langle X\rangle\right| \geq \epsilon \langle X\rangle\right) \geq \frac{1}{t} \cdot \frac{\sigma^2(X)}{\epsilon^2 \langle X\rangle^2} = \delta \checkmark \tag{47}$$

(b) [**2**] Suppose we only want a *weak estimate*: an estimate $\tilde{X}$ that is within $\epsilon \langle X\rangle$ with probability 2/3. Show that in this case we need only set $t = O(r^2/\epsilon^2)$.

Solution:

$$\delta = 1 - \frac{2}{3} = \frac{1}{3} \tag{48}$$

$$\frac{1}{3} = \frac{1}{t} \cdot \frac{\sigma^2(X)}{\epsilon^2 \langle X\rangle^2} \tag{49}$$

$$\Rightarrow t = 3\frac{r^2}{\epsilon^2} = \mathcal{O}\left(\frac{r^2}{\epsilon^2}\right) \checkmark \tag{50}$$

(c) [**8**] Now take the *median* of $O(\log(1/\delta))$ such weak estimates. Prove that this estimator does have the desired property of being within $\epsilon \langle X\rangle$ of the true estimate with probability $1 - \delta$. Note that we have done two things.

1. We've reduced the number of samples needed from $r^2/\epsilon^2\delta$ to $r^2 \log(1/\delta)/\epsilon^2$, which is an exponential reduction in dependence on $1/\delta$.
2. We've achieved a strong bound without using full Chernoff-like assumptions: we only needed X to have bounded variance.

Solution:

The median is represented in the above figure by $\langle X\rangle$ such that there are bounds $\langle X\rangle + \epsilon \langle X\rangle$ and $\langle X\rangle - \epsilon \langle X\rangle$ for how accurate we want our *weak estimates* to be. If it doesn't lie in the interval, then at least half of them lie outside the interval with probability $\left(\frac{1}{3}\right)^{\frac{t}{2}}$. The probability needs to be less than δ, so the upper bound on t can be calculated by

$$\left(\frac{1}{3}\right)^{\frac{t}{2}} = \delta \tag{51}$$

$$3^{-\frac{t}{2}} = \delta \tag{52}$$

$$t = 2\log\left(\frac{1}{\delta}\right) = \mathcal{O}\left(\log\left(\frac{1}{\delta}\right)\right) \checkmark \tag{53}$$

Question 6: Hashing with open addressing . . . . **[15]**

We've seen how to use random hash function to ensure only a small number of collisions in a hash bucket. We stored the overflow items in a *chain*, which is why we wanted to minimize its length. Such a hashing scheme is called *chaining*.

An alternate approach to hashing is called *open addressing*. Suppose we want to store n items in an array. We maintain an array of $2n$ slots, and also construct a hash function $h : U \times \mathbb{Z} \to [1 \ldots 2n]$. For each key k, the sequence $h(k, 0), h(k, 1), \ldots$ defines a *probe sequence* as follows.

- When k appear, the algorithm first tries to place it in $h(k, 0)$. If that entry is full, it tries $h(k, 1)$. If that is full, it tries $h(k, 2)$ and so on.
- When searching for an item q, we search the entries $h(q, i), i = 0, 1, \ldots$ until we either find q or find an empty spot (which means that q was never in the set).

Assume that $h(k, j)$ is uniform over $[1 \ldots 2n]$ for any k and that all $h(k, i)$ are independent.

(a) [**5**] Show that the probability an insertion takes more than r steps is at most 2^{-r}.

Solution:

$$P(\text{collision}) = \frac{1}{2n} \tag{54}$$

$$P(\text{collision after filling } r \text{ buckets}) = \left(\frac{1}{2n}\right)^r \tag{55}$$

$$P(\text{collision after filling } r+1 \text{ buckets}) = \left(\frac{1}{2n}\right)^{r+1} \tag{56}$$

$$P(\text{collision after} > r \text{ filled}) = \sum_{i=r}^{n} \left(\frac{1}{2n}\right)^i \tag{57}$$

$$= \sum_{i=o}^{n-r} \left(\frac{1}{2n}\right)^{i+r} = \underbrace{\left(\frac{1}{2n}\right)^r}_{<2^{-r}} \underbrace{\left[\frac{1 - \left(\frac{1}{2n}\right)^{(n-r+1)}}{1 - \left(\frac{1}{2n}\right)}\right]}_{<1} \tag{58}$$

$$\therefore P(\text{collision after} > r \text{ filled}) < 2^{-r} \checkmark \tag{59}$$

(b) [**5**] Show that for the i^{th} insertion ($i = 1, 2, \ldots, n$), the probability that more than $2 \log n$ probes are needed is $1/n^2$.

Solution:

Simply set r to $2 \log(n)$

$$2^{-2\log(n)} = \frac{1}{n^2} \tag{60}$$

(c) [**5**] Let X_i be the number of probes needed to insert item i, and let $X = \max X_i$. Show that $\Pr(X > 2\log n) \leq 1/n$.

Solution:

From the question, we know that we can set X to be the following

$$X = \max X_i \tag{61}$$

This leads to n different probabilities in it, but since it's the maximum X (which is n) then they are all the same probabilities, and you simply sum over all n of them

$$\sum_{i=1}^{n} P(X > 2\log(n)) = P(X > 2\log(n)) + P(X > 2\log(n)) + \cdots + P(X > 2\log(n)) \tag{62}$$

$$= n \cdot P(X > 2\log(n)) = n \cdot \frac{1}{n^2} = \frac{1}{n} \checkmark \tag{63}$$

Question 7: Derandomizing MAX CUT .... [**15**]

While randomness is an easy way to design an algorithm, random bits are an expensive resource, and we'd like to use as few of them as possible. The process of removing randomness from an algorithm is called *derandomization.* There are three basic approaches to derandomization: using weak randomness, eliminating randomess by brute force, and the method of conditional expectations. We will examine the latter two methods here.

(a) [**5**] Suppose we have a randomized algorithm that runs in time $T(n)$, uses $O(\log n)$ random bits and returns the minimum of some function with a probability of success greater than 2/3. Construct a deterministic algorithm that solves the same problem correctly in time $T'(n)$, expressed in terms of $T(n)$ and n (**Hint:** this is called the method of brute force).

The other approach works as follows. Suppose our randomized algorithm A produces some answer whose expectation is μ. We can imagine the algorithm as a sequence of deterministic operations punctuated by coin tosses $r_1, r_2, \ldots, r_m$. The value output by A is a function of the input and these coin tosses: ignoring the input, we can write the output value as $A(r_1, \ldots, r_m)$, and so $E[A(r_1, \ldots, r_m)] = \mu$

Consider the first coin toss r_1. We can write

$$E[A(r_1, \ldots, r_m)] = (1/2)E[A(0, r_2, \ldots, r_m)] + (1/2)E[A(1, r_2, \ldots, r_m)]$$

This is because the probability of $r_1 = 1$ is 1/2. Notice however that one of the two expectations $E_0 = E[A(0, r_2, \ldots, r_m)]$ and $E_1 = E[A(1, r_2, \ldots, r_m)]$ must be at least as large as μ (since $\mu = (E_0 + E_1)/2$). Suppose we can compute these two numbers. Then we can *deterministically* pick the value of r_1 and we are sure that the final answer we get is no worse than the expected value, while having used one less random bit.

Repeating this over and over again yields a final result in which all bits are picked deterministically, and yet the final answer is at least as large as μ. Thus, we've derandomized the algorithm without paying a penalty. The catch is of course the estimation of E_0 and E_1.

Now consider the randomized algorithm for MAX CUT that yields a 0.5-approximation: Pick a vertex and label it as being black or white with equal probability. Repeat for all vertices, and return the partition into black and white as the desired cut.

Solution: For a randomized algorithm, it takes in values such as `RA(X,r)` where `X` is some input and `r` is some random value. This can be turned into a deterministic algorithm `DA(X)` by the following algorithm

for $i = 1$ **to** n **do**

  if RA(X,i) $<$ min **then**
    min $=$ RA(X,i)
  end if
end for
return min

This yields a time complexity for the deterministic algorithm of

$$T'(n) = p(n) \cdot T(n)$$

Where $p(n)$ is some polynomial at least $\mathcal{O}(n)$

(b) [**10**] Derandomize this algorithm using the method of conditional expectations. What is the resulting deterministic algorithm? (**Hint:** think about how we might deterministically decide which color to give to a vertex based on the graph structure and how it influences E_0, E_1.)

Solution: An algorithm for deterministic coloring based off of the graph structure is:

1. Find the node with the highest degree. If there are multiple values, then choose one of them at random.
2. Color the given node one color, and color all the adjacent nodes the opposite color.
   - *For Example:* Color the highest degree vertex black and all connecting verticies white.
3. Find the highest degree that is not-colored and go back to step 2.

The analog to this in the max-cut problem is deleting all of the edges at those who are black, and leaving for all the white edges. Doing this will give the edges that need to be removed from the graph and give the Max Cut.