Homework 5 Solutions

1. Say that a CNF formula is 2-consistent if any two clauses can be simultaneously satisfied. Note that a CNF formula is **not** 2-consistent if and only if it contains the two clauses (x) and $(\bar{x})$ for some variable x. Also note that a CNF formula may be 2-consistent but unsatisfiable, as in $(x) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$.

   Show that for every 2-consistent CNF formula ϕ with m clauses there is always an assignment that satisifes at least $.61 \cdot m$ clauses.

   [Hint: give a probability distribution over assignments such that on average at least $.61 \cdot m$ clauses are satisfied]

   Solution:

   We will use the following distribution: for each clause with one variable of the form x, set x to be true with probability 0.61, and for each clause with one variable of the form $\overline{x}$, set x to be false with probability 0.61. Set all other variables to be true with probability 0.5. Since the formula is 2-consistent, we know that the clauses x and $\overline{x}$ cannot both exist, so the probability distribution described above is well-defined.

   Now let's see how many clauses we expect to satisfy under this distribution for an arbitrary 2-consistent formula ϕ. As usual, we can write the expected number of clauses satisfied as the sum of the probabilities that each clause is satisfied (why?).

   Each clause with one variable is satisfied with probability 0.61. For a clause with more than one variable, there are several cases, depending on the number of variables in the clause and whether the variables in the clause appear in single-variable clauses. It is clear, however, that the worst case will occur for clauses which have only two variables, both of which appear in the opposite form of single-variable clauses which appear elsewhere. In other words, for some x and y, the clause is $x \vee y$, and there are also the clauses $\overline{x}$ and $\overline{y}$ in ϕ. In this worst case, the probability that the clause is satisfied is $1 - 0.61^2 > 0.61$ (why?).

   Since for all clauses, the probability that the clause is satisfied is at least 0.61, we see that the expected number of clauses satisfied is at least $0.61m$. Finally, since the expectation is at least $0.61m$, there must exist an assignment which satisfies at least this many clauses.

   An alternative, worse, solution that would have gotten partial credit is as follows: let u be the number of unit clauses. One can always satisfy u clauses, because in a 2-consistent formula all unit clauses are satisfiable; also, a uniform random assignment satisfies on average at least $u/2 + 3(m-u)/4 = 3m/4 - u/4$ clauses. The best of the two strategies satisfies at least $.6m$ clauses, because either $u \geq .6m$ and then the former strategy is good, or $u < .6m$ and then the second strategy satisfies at least $3m/4 - u/4 > .6m$ clauses.

2. Pick at random a 3-CNF formula witn n variables and $6n$ clauses as follows. Start from an empty formula, then, for $6n$ times independently: pick at random a clause

among the $8\binom{n}{3}$ 3-CNF clauses that can be formed from n variables, add the clause to the formula. (We allow the same clause to appear more than once.)

Prove that the probability that the formula is satisfiable is exponentially small in n.

[Hint: compute the expected number of satisfying assignments, and use Markov inequality.]

Solution:

Let the random variable X be the number of satisfying assignments under the experiment described in this problem. Let X_i be the indicator random variable which is 1 exactly when assignment i (under some indexing of the assignments) satisfies the formula. Let X_i^j be the indicator random variable which is 1 exactly when clause j of the formula is satisfied by assignment i. Then $X = \sum_i X_i = \sum_i \left(\Pi_j X_i^j\right)$, and hence $\mathrm{E}[X] = E[\sum_i \left(\Pi_j X_i^j\right)] = \sum_i E[\left(\Pi_j X_i^j\right)] = \sum_i \left(\Pi_j E[X_i^j]\right)$, where the second equality is due to linearity of expectations, and the second equality is due to the fact that if random variables are independent, then the expectation of their product is the product of their expectations (why are the independent?).

Now $E[X_i^j] = \Pr[X_i^j = 1] = 7/8$, since for every choice of three variables to put in the clause, exactly one out of the 8 possible ways to write them in negated and non-negated form will be satisfied by the assignment i (this is true regardless of which assignment i is).

Putting this equality back into the original sum, we see that $\mathrm{E}[X] = \sum_i \left(\Pi_j \frac{7}{8}\right) = \sum_i \left(\frac{7}{8}\right)^{6n} = 2^n \left(\frac{7}{8}\right)^{6n} = \left(2\frac{7^6}{8^6}\right)^n$. Since $2\frac{7^6}{8^6} \approx 0.898 < 1$, the expected number of satisfying assignments is exponentially small in n. Finally, since $\Pr[X \geq 1] \leq \mathrm{E}[X]$ for a non-negative integer-valued random variable, we see that the probability that the formula is satisfiable is exponentially small in n.

3. Consider the following casino game. You pay a fixed stake of $\$\mu$, where $\mu = 3.5$. A fair 6-sided die is then tossed, and you receive $\$X$, where X is the score on the die. Since $\mathbf{E}[X] = 3.5 = \mu$ (see Lecture Note 1), your expected winnings are $\mathbf{E}[X] - \mu = 0$.

   Now suppose you play the game n times, and let X_i be the amount you receive on the ith play. Let $S_n = \sum_{i=1}^{n} X_i$, and $W_n = S_n - n\mu$ (your total winnings). What are $\mathbf{E}[W_n]$ and $\mathbf{Var}[W_n]$?

   Solution:

First let's compute $\mathrm{E}[W_n]$.

$$\begin{aligned}
\mathrm{E}[W_n] &= \mathrm{E}[S_n - n\mu] \\
&= \mathrm{E}[\sum_{i=1}^{n} X_i - n\mu] \\
&= \sum_{i=1}^{n} \mathrm{E}[X_i] - \mathrm{E}[n\mu] \\
&= n(3.5) - n\mu \\
&= 3.5n - 3.5n \\
&= 0.
\end{aligned}$$

Try to understand intuitively why this must be the answer.

Next let's compute $\mathbf{Var}[[]W_n]$.

$$\begin{aligned}
\mathbf{Var}[[]W_n] &= \mathrm{E}[W_n^2] - \mathrm{E}[W_n]^2 \\
&= \mathrm{E}[\left(\sum_{i=1}^{n} X_i - n\mu\right)^2] - 0^2 \\
&= \mathrm{E}[\sum_{i,j} X_i X_j - 2n\mu \sum_i X_i + n^2\mu^2] \\
&= \sum_{i,j} \mathrm{E}[X_i X_j] - 2n\mu \sum_i E[X_i] + n^2\mu^2 && (1) \\
&= \sum_i E[X_i^2] + \sum_{i \neq j} \mathrm{E}[X_i]\mathrm{E}[X_j] - 2n\mu \sum_i E[X_i] + n^2\mu^2 && (2) \\
&= n\frac{91}{6} + (n^2 - n)3.5^2 - 2n3.5(n3.5) + n^2 3.5^2 && (3) \\
&= (3.5^2 - 2 \cdot 3.5^2 + 3.5^2)n^2 + (\frac{91}{6} - 3.5^2)n \\
&= \frac{35}{12}n
\end{aligned}$$

Line (1) is due to linearity of expectation. Line (2) is due to the fact that any two rounds of the game are independent. The value 91/6 that appears in line (3) is due to the following calculation: $\mathrm{E}[X_i^2] = \sum_{i=1}^{6} \frac{1}{6} i^2 = \frac{91}{6}$.

This homework is due in class February 20, 2003

Homework 4

1. Andrew and Betty have a fair coin. They want to use it to generate a random sequence of $2n$ coin tosses containing exactly n heads and n tails.

   (a) Andrew suggests the following scheme: flip the coin $2n$ times; if you get exactly n heads, output the sequence; otherwise, try again. How many tosses do you expect to have to make under this scheme?

   [You may find question 3(b) of Homework 1 useful here.]

   Solution:

   Andrew's scheme involves repeatedly flipping $2n$ coins until exactly n heads and n tails are obtained. The probability of this happening is $p = \binom{2n}{n}\frac{1}{2^n} \sim \frac{1}{\sqrt{n\pi}}$, using Stirling's approximation as in question 3(b) of homework 1. The expected number of trials till we succeed, if the probability of success in each round is p, is once again $\frac{1}{p} \sim \sqrt{n\pi}$, giving a total of $2n\sqrt{n\pi}$ coin flips.

   (b) Betty claims that the following scheme is much more efficient: flip the coin until you have either n heads or n tails (one of these must happen before n tosses); output this sequence, padded at the end with tails or heads respectively to make the total length $2n$. Obviously this scheme requires at most $2n$ tosses. Is it a good scheme? Justify your answer with a precise calculation.

   [If you answer yes, you should show that all possible outputs are equally likely in Betty's scheme. If you answer no, you should show that there are sequences having different probabilities.]

   Solution:

   Betty's scheme involves tossing coins until either n heads or n tails have been obtained, and then padding the sequence to get $2n$ tosses with exactly n heads and n tails. To see that this does not give a truly random distribution, consider the sequence consisting of n heads followed by n tails. Since there are $\binom{2n}{n}$ valid sequences, this particular one should have probability $1/\binom{2n}{n} \approx \sqrt{n\pi}2^{-2n}$ using Stirling's approximation as in part (a). But in Betty's scheme, this sequence has probability 2^{-n} (i.e., the chance of getting 500 heads in a row), which is clearly way to high since $\sqrt{n\pi} \ll 2^n$.

   (c) Suggest a simple scheme for solving this problem that is better than both Andrew's and Betty's. How many tosses does your scheme require on average?

   Solution: Another scheme involves randomly selecting the positions of the n heads in the sequence of $2n$ tosses. One way to do this is to repeatedly choose random positions in the range 1 to $2n$ until n *distinct* random positions have been obtained. Let 2^m be the smallest power of two greater than $2n$. A random number in the range 1 to 2^m can be picked using m coin tosses (since $m =$

$\log(2^m)$). So say we're picking our k^{th} position, where $1 \leq k \leq n$. The probability that a random number in the range 1 to 2^m has not been previously selected and is $\leq 2n$ is $\frac{2n-k}{2^m} \geq \frac{2n-n}{4n} = \frac{1}{4}$, so the expected number of trials (each involving m coin tosses) till such a value is found is ≤ 4. The expected total number of coin tosses is then at most $\sum_{k=1}^{n} 4m \leq 4n(\log n + 1)$.

A second way to get $O(n \log n)$ is to do this: first write down n heads folled by n tails in a list. Now we will randomly shuffle the $2n$ elements of the list. To do this, start with the first element, and randomly choose another element in the list (including itself), and swap the positions of the two. Next, do the same with the second element and a randomly chosen one in positions $2 \ldots n$, and so on. After doing this for all n slots, the list will be in a random order. Since we used $O(\log n)$ coin flips to specify the swap in each step (why?), the total number of coin flips is $O(n \log n)$.

An even better method is the following. Suppose our first random coin toss is heads. Then for our next toss, we should use a biased coin with heads probability $(n-1)/(2n-1)$. (Why? Think about the conditional probabilities.) By the same argument, if the first i flips have h_i heads (and $i - h_i$ tails), then at the next step we should flip a coin with the probability of heads being $(n-h_i)/(2n-i)$, and so on. But how do we simulate these biased coins with our fair coin? We claim that *any* bias p can be realized using an expected number of only <u>two</u> fair coin flips!!! To see this, write $p = 0.p_1p_2\ldots$ in binary. Now generate a random number $r = r_1r_2\ldots$ between 0 and 1 by successively choosing each binary digit r_i using an independent fair coin flip. We can stop when we know that $r > p$ or that $r < p$ (corresponding to an outcome of heads or tails respectively for our biased coin). And when do we know this? It is when we reach the first i for which $r_i \neq p_i$. But the expected number of tosses for this event to happen is just 2. (Why?) Putting all this together, the expected number of coin flips needed to generate the entire random sequence of $2n$ flips is only $4n$.

2. This problem concerns Karger's randomized algorithm for MinCut (see Note 4).

   (a) Suppose that Karger's algorithm is applied to a tree G. Show that it finds a minimum cut in G with probability 1.

   Solution:

   Since Karger's algorithm always contracts edges, at any stage of the algorithm the graph must be a tree. Hence it is a single edge in the last step, which is a minimum cut of a tree.

   (b) Suppose we modify Karger's algorithm as follows: instead of choosing an edge uniformly at random and merging its endpoints, the algorithm chooses *any* two distinct vertices uniformly at random and merges them. Show that for any n there is a graph G_n with n vertices such that when the modified algorithm is run on G_n, the probability that it finds a minimum cut is *exponentially* small in n.

   Solution:

   Consider a graph which is composed of two cliques of size $n/2$ plus one edge between the cliques. The modified algorithm will not be able to find the minimum cut with better than an exponentially small probability for this graph.

To see this, notice first of all that the algorithm will fail if it chooses a pair of vertices with one in each clique at any step. Now consider the first $n/4$ steps of the algorithm on this graph. We will show that the probability that it does not choose such a pair is exponentially small in n. Since the algorithm has already such a small probability of success by this stage, it must have an even lower probability of success overall.

If the algorithm has not yet failed (chosen such a pair of vertices) by stage $i+1$, one clique will have $n/2-k$ vertices and the other one will have $n/2-(i-k)$, for some $k \in \{0,\ldots,i\}$. We claim that if $k=0$ or $k=i$, the probability of not failing in this stage is maximized (to prove this, write down the probability of not failing as a function of k, and show that it achieves its maximum for those values of k). Hence we can upper-bound the probability of not failing at this step by the probability when one clique has size $n/2$ and the other has size $n/2-i$. In this case, the probability of not failing is $\frac{\binom{n/2}{2}+\binom{n/2-i}{2}}{\binom{n/2+n/2-i}{2}}$; if $i \leq n/4$, a simple calculation shows that it is at most $2/3$. Hence the probability of success at stage i, given success at all previous stages, is at most $2/3$.

Hence the probability of success through the first $n/4$ stages is at most

$$\left(\frac{2}{3}\right)^{n/4} = \frac{1}{\frac{3}{2}^{n/4}} = \frac{1}{\left((3/2)^{1/4}\right)^n}$$

which is exponentially small in n.

(c) How many times would you have to repeat the modified algorithm of part (2b) in order to have a reasonable chance of finding a minimum cut? What does this imply about the practical utility of the modified algorithm?

Solution:

Using the usual coin tossing argument, we would have to repeat the algorithm an exponential number of times in order to have a reasonable chance of finding a minimum cut, which implies that this algorithm is not practical at all.

(d) Show that for every $n \geq 3$ there is a graph G_n with n vertices that has $n(n-1)/2$ distinct minimum cuts.

Solution:

A cycle on n vertices is such a graph, since every pair of edges is a minimum cut, and there are $\binom{n}{2} = n(n-1)/2$ such pairs.