CS 173: Discrete Mathematical Structures, Spring 2009
Homework 9 Solutions

---

1. **[10 points] Pigeonhole Principle**
   Let S be a set of ten distinct integers between 1 and 50. (Distinct means that no two elements of S are the same.) Use the pigeonhole principle to show that there are two different 5-element subsets of S with the same sum.

   The "sum" of a set of numbers is what you get if you add up all the numbers in the set. For example, suppose that $S = \{1, 2, 3, 4, 5, 30, 31, 32, 33, 34\}$. Consider the two subsets $\{30, 2, 3, 4, 5\}$ and $\{31, 1, 3, 4, 5\}$. The sum of the first subset is $30 + 2 + 3 + 4 + 5 = 44$, which is the same as the sum of the second subset, i.e. $31+1+3+4+5$. So these two subsets are different but they have the same sum. You need to show that this works no matter how we choose the ten elements of S.

   Solution: The number of 5-element subsets of S is $\binom{10}{5} = 252$. An upper-bound on the number of distinct sums of 5-element subsets is $5 * 50 = 250$, since each number in a subset can equal no more than 50.

   Since there are more 5-element subsets of S than there are distinct sums of 5-element subsets, by the pigeonhole principle, there must exist two 5-element subsets of S with the same sum.

2. **[10 points] Graph Theory and Degree Sequences**
   The degree sequence for a graph G is the sequence of the degrees of the vertices in the graph arranged in non-increasing order. For example the degree sequence for W_4 would be 4,3,3,3,3.

   (a) Let m and n be positive integers with $n > m$. What is the degree sequence for the graph $K_{m,n}$?

   Solution: $K_{m,n}$ is the complete bipartite graph with vertices partitioned into sets of size m and n.
   Since $n > m$, $K_{m,n}$ has degree sequence $x_1, ..., x_m, y_1, ..., y_n$ where for all $1 \leq i \leq m$, $x_i = n$ and for all $1 \leq i \leq n$, $y_i = m$.

   (b) What simple graph with n vertices has a degree sequence of $n-1, n-1, ..., n-1$?

   Solution: K_n has such a degree sequence, as each node in K_n is connected to each of the other $n - 1$ nodes.

   (c) Consider a full binary tree T with n vertices. Suppose there are i internal vertices. What would the degree sequence for T look like?

   Solution: $x_1, ..., x_i, x_{i+1}..., x_n$ where for all $1 \leq j \leq i$, $x_j = 3$, and $x_{i+1} = 2$, and for all $i + 1 < j \leq n$, $x_j = 1$.

This is because the root vertex has degree 2, the i internal vertices each have degree 3, and the $n - i - 1$ leaves each have degree 1.

(d) A sequence $d_1, d_2, ..., d_n$ is called *graphic* if it could be the degree sequence of a simple graph. Is the sequence $6, 5, 4, 3, 2, 1$ graphic? Explain your answer.

Solution 1: No, the sequence $6, 5, 4, 3, 2, 1$ is not graphic. By the Handshaking Theorem, no graph has an odd number of vertices of odd degree (in this case, three vertices of degree 5, 3, and 1.)
Solution 2: No, the sequence $6, 5, 4, 3, 2, 1$ is not graphic. In a graph with such a degree sequence, the node with degree 6 would need to be connected to 6 other nodes, but there would be only 5 other nodes. Again, recall that a simple graph does not contain self-loops or multiple edges with the same endpoints.

3. **[10 points] Conditional Probability and Independence**

(a) Let E be the event that the bit string of length 5 contains an odd number of 1s. Let F be the event that the string ends with a 0. Are E and F independent events?

Solution: Yes, E and F are independent events.
The independence of E and F is not immediate, since they both depend on the value of the last bit. Therefore we use the formal definition of independence and show that $p(E \cap F) = p(E)p(F)$.
We first note that $p(E) = p(F) = 1/2$.
We next determine $p(E \cap F)$, observing that $E \cap F$ occurs when a bit string of length 5 contains an odd number of 1s and ends in a 0. This occurs when the last bit is 0 (probability $1/2$) and the first 4 bits contain an odd number of 1s (probability $1/2$). Since these events refer to different bits in the string, they are clearly independent, so $p(E \cap F) = 1/2 * 1/2 = 1/4$.
Thus $p(E \cap F) = 1/4 = 1/2 * 1/2 = p(E) * p(F)$.

(b) What is the conditional probability that exactly 4 heads appear when a fair coin is flipped 5 times, given the first flip came up tails?

Solution: Exactly 4 heads will appears out of the 5 flips if and only if the remaining 4 flips all come up heads. The probability of this is $(1/2)^4 = 1/16$.
One can also compute the answer by labeling E the event that exactly four heads appear out of 5 flips, and labeling F the event that the first coin flip comes up tails. Then the probability that a total of 4 heads will appear given that the first flip came up tails is $p(E|F) = \frac{p(E \cap F)}{p(F)} = \frac{(1/2)^5}{1/2} = (1/2)^4 = 1/16$.
(Here $p(E \cap F)$ is the probability that the first coin flip is tails and a total of 4 heads are flipped.)

4. **[10 points] Expectation**

Imagine there are N couples at a party, and suppose m people get sleepy and have to go home. When a person goes home, his or her date has to leave with them. We need to find the expected number of couples left at the party.

(a) First, let's consider a single couple, couple i. Define a random variable X_i to be:

$$X_i = \begin{cases} 1 & \text{if couple } i \text{ stays} \\ 0 & \text{if couple } i \text{ leaves} \end{cases}$$

This sort of function is known as an *indicator*.

What is the expected value of X_i? Clearly explain your answer.

Solution 1: The couple stays only if neither gets sleepy. To compute this probability, we count the number of ways to choose m sleepy people from everyone else excluding this couple, then divide by the total number of ways to choose m sleepy people. So we get:

$$p = \frac{\binom{2N-2}{m}}{\binom{2N}{m}} = \frac{\frac{(2N-2)!}{(2N-2-m)!m!}}{\frac{(2N)!}{(2N-m)!m!}} = \frac{(2N-m)(2N-m-1)}{2N(2N-1)}$$

The expected value of X_i is thus $\mathbf{E}(X_i) = p * 1 + (1-p) * 0 = p$.

Solution 2: The couple stays only if neither gets sleepy, which occurs with probability $p = (1 - \frac{m}{2N})(1 - \frac{m}{2N-1})$. This is because $\frac{m}{2N}$ is the probability that the first person in the couple gets sleepy, and $1 - \frac{m}{2N}$ is the probability that the first person in the couple does *not* get sleepy and stays.

$1 - \frac{m}{2N-1}$ is the probability that the *second* person in the couple does not get sleepy and stays, *given* that the first person does not get sleepy and stays. The denominator is only $2N-1$ since one person is already known to not be sleepy. If you fiddle with the algebra, you'll see that this equation is equal to the one in the previous solution.

Again, the expected value of X_i is thus $\mathbf{E}(X_i) = p * 1 + (1-p) * 0 = p$.

(b) What is the expected number of couples left at the party? Your answer should be a function of the variables N and m.

Hint: Suppose we have n random variables X_i and we wish to compute the expected value of $\sum_{i=1}^{n} X_i$. This means we have sum of n functions and we wish to find the expected value of the sum. We can do this by using the *linearity of expectation* which establishes that $\mathbf{E}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbf{E}(X_i)$. This means that we can find the answer by computing the expectation for each X_i separately and then summing the results.

Solution 1: We use our result for **Solution 1** of part a. By linearity of expectation, the expected number of couples left at the party is $\sum_{i=1}^{N} \mathbf{E}(X_i) = \sum_{i=1}^{N} p = N * p$ which is equal to:

$$\frac{(2N-m)(2N-m-1)}{2(2N-1)}$$

Solution 2: We use our result for **Solution 2** of part a. By linearity of expectation, the expected number of couples left at the party is $\sum_{i=1}^{N} \mathbf{E}(X_i) = \sum_{i=1}^{N} p = N * p$ which is equal to:

$$N\left(1 - \frac{m}{2N}\right)\left(1 - \frac{m}{2N-1}\right)$$

Note: Algebraic manipulation shows that the two solutions are equivalent.

5. **[10 points] An Algorithm**

Consider the problem of writing a program to verify polynomial identities of the form:

$$(a_1x + a_2y)^n = b_0x^n + b_1x^{n-1}y^1 + ... + b_{n-1}x^1y^{n-1} + b_ny^n$$

where n is a positive integer and the a_i and b_i are real numbers.

One way to verify the identity is to use the binomial theorem to find the coefficients generated by the left hand side and see if they match the b_i's on the right hand side. Consider the following pseudo-code procedures to do that:

```
procedure factorial(k)
fac := 1
for i := 1 to k
    fac := fac · i
return fac
```

```
procedure VerifyBinomial( a1, a2,b0..., bn)
matches := true
for i := 0 to n
begin
    a1pow := 1
    for j := 1 to n − i
        a1pow := a1pow · a1
    a2pow := 1
    for j := 1 to i
        a2pow := a2pow · a2
    c := factorial(n)/(factorial(n − i) · factorial(i))
    c := c · a1pow · a2pow
    if(c ≠ bi)then
        matches := false
end
return matches
```

(a) State a big-Θ bound on the number of multiplications done by the procedure **factorial** in terms of the input k.

Solution: In factorial, the loop is iterated k times, with one multiplication per iteration, resulting in $k = \Theta(k)$ multiplications.

(b) State a big-Θ bound on the number of multiplications done by the procedure **VerifyBinomial** in terms of the input n. Use your answer from part (a) when considering how many multiplications **factorial** does.

Solution:

Consider a single iteration of the outer loop in VerifyBinomial:

the first inner loop is iterated $n - i$ times and the second inner loop is iterated i times, for a total of n multiplications.

Then an intermediate value for c is computed, causing $n + (n - i) + i + 1 = 2n + 1$ multiplications to be performed (recalling our result from part a.)

Finally, the final value of c is computed, resulting in two more multiplications.

Thus, each iteration of the outer loop results is $3n + 3$ multiplications. The outer loop is executed $n + 1$ times, resulting in $(3n + 3)(n + 1) = \Theta(n^2)$ total multiplications.