Homework 6

Discrete Structures
CS 173 [B] : Fall 2015

Released: Fri Apr 10
Due: Fri Apr 17, 5:00 PM

Submit on Moodle.

PART 1 (Machine-Graded Problems) on Moodle. [25 points]

PART 2 [75 points]

1. **Recurrence Relation** [20 points]

   Recall that $\binom{n}{k}$ is the number of subsets of size k that a set of size n has.

   (a) Use mathematical induction to prove that, for all $n, k \in \mathbb{N}$ such that $k \leq n$, we have $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, based on the following: $\forall n \in \mathbb{N}$, $\binom{n}{0} = \binom{n}{n} = 1$; and, for $n \geq 1$, $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (which we obtained by considering separately the subsets of size k that contain and do not contain a fixed element from the set).

   (b) Above, $\binom{n}{k}$ was expressed in terms of $\binom{n-1}{i}$ for two different values of i. Use a similar argument to express $\binom{n}{k}$ in terms of $\binom{n-2}{i}$ for different values of i (for $n \geq 2$).

   [Hint: Alternately, note that $\binom{n}{k}$ is the coefficient of x^k in the expansion of $(1+x)^n = (1+x)^2 \cdot (1+x)^{n-2}$.]

2. **Partitions from Onto Functions.** [20 points]

   Consider the following definitions.

   - For a function $f : A \to B$, let $\hat{f} : A \to \text{Image}(f)$ be the unique *onto function* such that $\forall x \in A$ $f(x) = \hat{f}(x)$.
   - For a function $g : A \to C$, let the pre-image function $PI_g : C \to \mathbb{P}(A)$ be defined by $PI_g(y) = \{x \mid f(x) = y\}$.
   - For a function $f : A \to B$, let the "pre-image partition" of A, be defined as $PP_f = \text{Image}(PI_{\hat{f}})$.
   - Define an equivalence relation $\sim$ between functions $f_1 : A \to B$ and $f_2 : A \to B$ as follows: $f_1 \sim f_2$ if $PP_{f_1} = PP_{f_2}$.

   Answer the following with respect to the above definitions.

   (a) Suppose $A = \{a, b, c\}$ and $B = \{1, 2, 3\}$. Consider $f : A \to B$ defined as $f(a) = f(b) = 1$ and $f(c) = 2$. Also, let $f' : A \to B$ be defined as $f'(a) = f'(b) = 3$ and $f'(c) = 2$

      i. Describe the functions $\hat{f}$ and $\hat{f}'$.

      ii. Describe the functions $PI_{\hat{f}}$ and $PI_{\hat{f}'}$.

iii. Describe the partitions PP_f and $PP_{f'}$.

(b) Let $f : A \to B$, where $|A| = n$, $|B| = k$ and $|\text{Image}(f)| = i$. Then how many functions f' are there such that $f \sim f'$? Justify your answer.

3. **Lottery** [20 points]

Counting is intimately connected to computing the *probability* of various events. In this problem we shall use counting to calculate the probability of winning lotteries.

In a certain kind of lottery, each player submits a sequence of n digits (between 0 and 9). A player wins a grand prize if her submission exactly matches a sequence of n digits selected by a random mechanical process. She wins a smaller prize if only $n-1$ digits are matched (e.g., for $n = 4$, if the submission is 1248 but the machine chooses 1298, then a small prize is awarded).

(a) How many ways can the mechanical process choose a sequence of n digits? Use this to compute the probability of a player (who has submitted a single sequence) winning the large prize, assuming that the mechanical process chooses each possible sequence equally likely (i.e., uniformly at random).

[Hint: You can use the following fact regarding probability. If one item is chosen out of N possible items uniformly at random, then the probability of it being any priori fixed item is $1/N$.]

(b) For any sequence of n digits that a player picks, how many sequences are there which, if chosen by the mechanical process, would result in the player winning a small prize? Use this to compute the probability that a player (who has submitted a single sequence) wins the small prize.

[Hint: The probability in this case is $\frac{p}{N}$, where p is the number of sequences, which if chosen by the mechanical process, leads to a small prize, and N is the total number of all possible sequences that the mechanical process can choose.]

4. **Sorted Strings** [15 points]

Consider strings made up of lowercase letters, `a-z`. We say that a string is a "sorted string" if the letters in it appear in alphabetic order. For instance, `bbn` and `tux` are sorted strings, but `ibm` is not.

(a) How many sorted strings of length 3 are there?

[Hint: Can you relate a sorted string to a multi-set?]

(b) How many sorted strings of length 3 are there in which no letter repeats? (Thus `bbn` should not be counted, but `tux` should be.)