Review Exercises

1. ($\star$) What is the runtime of isPrime? Is it polynomial?
2. Prove that a number is divisible by 3 if and only if its digit sum is divisible by 3.
3. Write a function isOdd that determines if an int is odd.
4. What is 5%3? What is $-7\%2$? What is $7\% - 2$?
5. Suppose you have an int a and a mod M. Determine how to get the smallest non-negative value b such that M divides $a - b$. That is, we want the smallest non-negative b with $a \equiv b \pmod{M}$.
6. Let $a = qb + r$ where $a, b, q, r \in \mathbb{Z}$ are non-negative and $0 \leq r \leq b$. Show that if g divides r and b then g divides a. What does this allow?

Solutions

1. Roughly $\sqrt{n}/\log n$. This is polynomial in n, but not polynomial in the number of bits in n. BigInteger has a probabilistic prime function whose runtime is dependent on the probability of correctness.
2. Consider the number $\sum_{k=0}^{n} d_k 10^k$ where $0 \leq d_k \leq 9$. Then mod 3 this is just $\sum d_k$ as $10^k \equiv 1$ mod 3.
3. **boolean** isOdd(**int** a){ **return** a%2!=0; } Note that a%2==1 is incorrect.
4. $5\%3 = 2$. Note that $-7/2 = -3$ so

   $$-7 = (-7/2) \cdot 2 + r = -6 + r \implies r = -1.$$

   Furthermore

   $$7 = (7/-2) \cdot -2 + r = 6 + r \implies r = 1.$$

   Hence $-7\%2 = -1$ and $7\% - 2 = 1$.
5. You could do b = ((a % M)+ M)% M;, but the following avoids doing 2 mods:

   ```
   int b = a % M;
   if (b < 0) b += M;
   ```

6. If $r = gj$ and $b = gk$ then $a = qb + r = qgk + gj = g(qk + j)$. This allows us to use the Euclidean algorithm for finding GCDs:

   ```
   int gcd(int a, int b) { return b == 0 ? a : gcd(b,a%b); }
   ```

Discrete Probability and Combinatorics

Many problems involving probability can be reduced to combinatorics by simply counting the number of ways some phenomenon can occur. In the following problems, we review some of these techniques.

Combinatorics Review

1. Show how to compute the number of subsets of $\{1, 2, \ldots, n\}$ of size k. How would you compute the formula using code?

2. ($\star$) Let A, B be sets with $|A| = 7$ and $|B| = 3$. How many functions $f : A \to B$ are there? How many are onto? How many are one-to-one?

3. How many solutions are there to the equation $a + b + c = 10$ where a, b, c are non-negative integers? What if they are positive?

4. ($\star$) Given a complete graph with $2n$ vertices, how many distinct perfect matchings are there? Equivalently, given the numbers $\{1, 2, \ldots, 2n\}$, how many distinct ways are there to pair the numbers up?

5. ($\star\star$) Give an algorithm to determine how many permutations of n numbers are there such that no number is fixed? Such a permutation is called a derangement.

Solutions

1. $\binom{n}{k}$

```
long ret = 1;
for (int i = 0; i < k; ++i)
{
  ret *= n-i;
  ret /= 1+i;
}
```

2. 3^7 functions, none are one-to-one. The number that are onto is (using inclusion-exclusion)

$$3^7 - 3 \cdot 2^7 + 3 \cdot 1^7.$$

More generally,

$$\left|\bigcup A_n\right| = \sum |A_n| - \sum_{i<j} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k| - \cdots .$$

3. $\binom{10+2}{2}$, $\binom{7+2}{2}$.

4. $\frac{(2n)!}{n!2^n}$.

5. Recurrence $D_n = (n-1)(D_{n-1} + D_{n-2})$ as we can either pair n with any number in a 2-cycle, or map n to a number k, and map whoever was mapped to k to n. Conversely, every such derangement is achievable in this fashion.

   Alternatively, by inclusion exclusion,

$$D_n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}.$$

   To see this, note that there are $n!$ total permutations, and then $n \cdot (n-1)!$, and $\binom{n}{2}(n-2)!$ are the next two terms in the inclusion exclusion sum (permutations that fix 1 element, permutations that fix 2 elements). This shows the limit is $n!/e$.

Discrete Probability and Expectation

When discussing probability on discrete sets, we will assign a non-negative probability to each element of a set such that the probabilities sum to one. The most common assigment for a finite set, is to assign equal probabilities to each item (more formally, we are assigning probabilities to singletons, but we will ignore this distinction).

Consider the problem of choosing a letter randomly from the English alphabet. We can ask what is the probability of selecting a vowel. The answer is 5/26 as there are 5 vowels each occurring with probability 1/26, and the probability of a disjoint union is the sum of the probabilities. We can also ask the same question, but restricting to the first 5 letters of the alphabet. Then the answer is 2/5. This immediately gives us a definition for conditional probability: $\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}$.

A random variable is a function that assigns real numbers to each element in our probability space (in both the discrete and continuous cases above). Technically, we require a random variable to be measurable, but we will ignore this formality. We will look at two types of random variables. For discrete random variables, we know the probability (the probability mass function) that the random variables attains each value (like throwing a die, or a student taking an exam), and these probabilities sum to one.

Given a random variable, we can compute its expectation. For a discrete random variable, let $x_1, x_2, \ldots$ be the set of values it attains. Then we have

$$E[X] = \sum_i x_i \Pr(X = x_i).$$

We can also compute conditional expectations using conditional probabilities (conditioning on an event A):

$$E[X|A] = \sum_i x_i \Pr(X = x_i|A).$$

If you have another random variable Y then you can say things like

$$f(y) = E[X|Y = y] = \sum_i x_i \Pr(X = x_i|Y = y),$$

where our event is $Y = y$. As this is a function of the value y, we can compute a compose it with the random variable Y to get the random variable $f(Y)$ which we denote $E[X|Y]$. We have results for expectation like linearity $E[aX + bY] = aE[X] + bE[Y]$ and the law of total expectation $E[E[X|Y]] = E[X]$

Probability Exercises

1. What is the probability of getting dealt a full house (standard 52-card deck; full house means a 3-of-a-kind and a 2-of-a-kind)?
2. Give a formula for the probability that 2 people have the same birthday in a class of 38 (assume a year has 365 days)?
3. How many people are required to guarantee 7 people have the same birthday?
4. Given a fair coin, how can you simulate a 6 sided die? What is the expected number of flips to do this?
5. Give the expected number of 6-sided die rolls you must make before you get:
   (a) ($\star$) 2 fours in a row?
   (b) 2 fours total?
6. I have a sack with 12 quarters: 4 are normal, 4 are double-headed, 4 are double-tailed. I randomly pick a quarter and randomly show you a side. If you see a head, what is the probability that I also see a head?
7. ($\star\star$) Suppose 10 equally skilled football teams play each other twice during the season (90 games), and each game is won or lost giving the winner a point. You want to compute the probability that exactly two teams end up tied for first by simulating the season over and over again. How many times should you run the simulation to feel confident in your answer to 3 digits of precision?

Solutions

1. $$\frac{13 \cdot 12 \cdot \binom{4}{3}\binom{4}{2}}{\binom{52}{5}}.$$
2. $$1 - \frac{365 \cdot 364 \cdots 329 \cdot 328}{365^{38}} \approx .864.$$

3. $6 \cdot 365 + 1$.

4. 3 flips gives 8 possible values (0-7). Reflip on 6 or 7 (i.e., first two flips heads). Expectation is

$$E(X) = 2 + .25E(X) + .75(1) \implies .75E(X) = 2.75 \implies E(X) = 11/3.$$

5. (a)

$$E[R_0] = 1 + 5E[R_0]/6 + E[R_1]/6, E[R_1] = 1 + 5/6E[R_0] \implies E[R_0] = 42.$$

   (b) For 1 four we have $E[R] = 1 + 5/6E[R] \implies E[R] = 6$. Thus the answer is 12.

6. Of 12 sides you could be looking at, 8 have me looking at a head. Hence the probability is 2/3.

7. The mean of your Bernoulli trials is approximately distributed as $N(p, p(1-p)/n)$. That is, the standard deviation behaves like

$$\sigma = \sqrt{\frac{p(1-p)}{n}}.$$

   If you want a 95% confidence interval accurate to 3 digits, we need

$$2\sigma < .0005 \implies \frac{2\sqrt{p(1-p)}}{\sqrt{n}} < .0005 \implies n > (4000\sqrt{p(1-p)})^2 \approx 4 \cdot 10^6.$$

Markov Chains

A finite Markov Chain is essentially a finite state machine where the edges are labeled with probabilities. The labels on all edges leaving a node sum to one. Conditional on being on a given state, the edges give the probability that you will be on each of the adjacent nodes on the next step. Given a vector of probabilities that you are on each state, you can compute the probabilities after one step by taking each possible step.

Markov Chain Exercises

1. You have 60 health and your opponent has 80 health. Each turn you roll a 10-sided die, and he rolls a 6-sided die to determine damage inflicted (i.e., how much to decrease your opponent's health by). The battle ends when at least one player has lost all health (if your health goes negative, it just becomes 0). Give an algorithm to compute the probability you win with at least 5 health remaining. Show how to compute your expected remaining health.

2. You and your opponent start at spot 0 on a circular track of length 10 (spots 0-9). Each time you both roll a 6-sided die and advance an according number of spots. A player finishes if they land exactly on spot 0. Show how to compute

(a) the probability you finish while your opponent is on spot 7.

(b) ($\star$) the probability you finish at least three rolls before your opponent.

Solutions

1. Make a 61×81 table representing the probability of each possible health pair. Simulate until all probability lie in end states (with one of your healths at zero). Add up all end states where your health is at least 5. To compute your expected remaining health, sum over all end states multiplied by your health value at each state.

2. (a) Make a 10×10 board. Keep iterating the game, removing probabilities corresponding to the game ending (accumulating the ones where your opponent is on spot 7). Each step the total probability remaining on the board decreases. Eventually the remaining probability will be negligible.

   Alternatively, you could construct a linear system where each variable has the from $W_{i,j}$ denoting the probability of eventually satisfying the condition given you are on spot i and your opponent is on spot j. Solving this linear system gives the result.

   (b) Same as the previous problem, but duplicate the end state 3 times. After you reach spot zero, the next step you are on the second spot zero state, and then the third spot zero state. You satisfy the condition when you are on the third spot zero state, and the opponent hasn't reached spot zero yet.

Game Theory

The game theory we will look at comes down to a few techniques: expectation maximization, minimax, working backwards, and nim games. In expectation maximization, you are given a game to play, you are allowed to make choices, and you must determine the expected value of playing the game assuming optimal choices are made (like the coin flipping extra credit problem).

In minimax, we typically look at games with two players, and you must determine if one player can win assuming perfect play. We can compute this using recursion/memoization on the state of the game, assuming each player tries to make their best move maximizing their score, or chances of winning.

The concept of working backwards, is that many game theory problems are easily solved in simple cases when you are near the end of the game. Then, using induction, we can often work backwards and determine the solution for more complex cases.

Finally, the concept of nim games will be explored in the exercises.

Game Theory Exercises

1. Ten pirates (numbered 1-10) have just found 100 gold pieces. The lowest number pirate suggests an allocation of the gold pieces. If more than half of the pirates disagree, they

throw the lowest numbered pirate overboard and repeat the process. Assuming you are pirate 1, what is the maximum amount of gold you can receive without swimming? All pirates are rational and want to maximize their gold. They will disagree unless it strictly benefits them.

2. Given a partially complete 4-by-4 tic-tac-toe game where a player must achieve 4 in a row, determine if the player to move can guarantee a win given optimal play.

3. ($\star\star$) You have 10 piles of stones, the ith pile holds s_i stones ($1 \leq s_i \leq 10^9$). Each turn you or your opponent (turns alternate) can remove any number of stones from one of the piles. The first player that cannot move loses. Give an algorithm to determine if you can win with optimal play.

4. ($\star$) You are playing a game in which four fair coins are flipped and the amount of money you receive in dollars is equal to the number of heads that appear in total. If you do not like the outcome of the first four flips, you have the option to re-flip the four coins, but you are obligated to take the second outcome. Determine the expected value of this game.

5. ($\star\star$) You are playing a game where the player gets to draw the number 1-100 out of hat, replace and redraw as many times as they want, with their final number being how many dollars they win from the game. Each redraw costs one dollar. How much would you charge someone to play this game?

Solutions

1. Let's go backwards.

   (a) If there is 1 pirate, he gets all 100 pieces.

   (b) If there are 2 pirates, number 1 gets all 100, as even if pirate 2 disagrees, he will never be more than half the pirates.

   (c) If there are 3 pirates, number 1 gets 99 and gives 1 gold piece to pirate 3.

   (d) If there are 4 pirates, number 1 gets 99 and gives 1 gold to pirate 3.

   (e) If there are 5 pirates, numbers 1 gets 98 and gives 1 gold to pirates 3 and 5.

   (f) In general, if there are n pirates, 3, 5, 7, etc. all get 1 gold, and pirate 1 takes the rest. So for 10 pirates, pirate 1 gets 96 gold.

2. Firstly, denote a draw as the first player losing (as it means he hasn't won). For every board, the current player either can force a win with perfect play, or is forced to lose (i.e., every move loses) with perfect play. This can be shown by induction. Write a memoized function that takes the current board, and determines if the player to move can force a win by recursing on all moves (you can back out whose turn it is from the board).

3. Compute the xor of all of the piles. If the number is non-zero, you can force a win. Otherwise, you lose with perfect play. The winning strategy is to always make a move that returns the xor to zero for the other player. To find your move, find the highest order bit that is on in the xor. Play in a pile that has that bit on, and move to set the xor to zero.

4. The number of heads H satisfies

   $$\Pr(H=0)=\frac{1}{16}=\Pr(H=4), \Pr(H=1)=\frac{1}{16}\binom{4}{1}=\Pr(H=3), \Pr(H=2)=\frac{1}{16}\binom{4}{2}.$$

   The expectation is

   $$4\cdot\Pr(H=4)+3\cdot\Pr(H=3)+2\cdot(1-\Pr(H=3)-\Pr(H=4))=\frac{1}{4}+\frac{3}{4}+\frac{22}{16}=\frac{38}{16}.$$

5. Extra Credit.