Bootcamp Ch06

Weixiang Ding Andrew Id: weixiand

6.1 Take one pill from bottle 1, two pills from bottle 2 … and 20 pills from bottle 20. Since only one bottle has pills of weight 1.1 grams, we can get the number of the bottle from the equation $n = \frac{Total\ weights - 210}{0.1}$. In this problem, we use the number of pills of each bottle as their index.

6.2 As provided in the problem, the probability of winning game one is p. For game two, there are two conditions to win that making two shots or making three shots. If making two shots, the probability should be $p * p * (1-p) + p * (1-p) * p + (1-p) * p * p = 3p^2(1-p)$. If making three shots, the probability should be $p * p * p = p^3$. To win the second game, the probability is $3p^2(1-p) + p^3 = 3p^2 - 2p^3$. Let $p = 3p^2 - 2p^3$, we can obtain that $p = 1$ or $p = 0.5$. Therefore, if we choose p to be 1 or 0.5, these two games have the same probability to win. If we choose p to be smaller than 0.5, game one is more likely to win and if we choose p to be larger than 0.5, game two is more likely to win.

6.3 No, I can't. **Proof.** Since this is a 8*8 chessboard, let's assume that the color of each square is only black or white and it is different from its adjacent square. The two diagonally opposite corners must be the same color there are 13 squares between them. Assume that they are all white. Therefore we have 30 white squares and 32 black squares. Since each domino can only cover two squares and these two squares have different colors, there are two black squares that cannot be covered. □

6.4 Since all the ants walk at the same speed, if they walk in the direction of clockwise or counter clockwise, whey will not make collisions. Each ant has two states, so there are totally 2^3 states for three ants. And there are only two states to make collisions. Therefore, the probability is $1 - 2 \div 2^3 = 0.75$. For n ants on an n-vertex polygon, the process is similar and the answer should be $1 - 2 \div 2^n = 1 - \frac{1}{2^{n-1}}$.

6.5 First we fill the three-quart jug with water, and then pour all the water from the three-quart jug to the five-quart jug. After we repeat this process and to fill the five-quart jug. Therefore we have one quart in the three-quart jug and five quarts in the five-quart jug. Then we pour all the water of the five-quart jug to the ground and transfer the one quart water from the three-quart jug to the five-quart jug. After that, we refill the three-quart jug and transfer all the water to the five-quart jug. Finally we get four quarts water in the five-quart jug.

Five-quart	Three-quart
0	3
5	1
0	1
1	3
4	0

6.6 Assume that there are n people in the island and *x* people with blue-eye. If *x*=1, the only one will take the flight at that evening because he know others are not blue-eye and at least one person is blue-eye. If *x*=2, each of these two people are unsure whether himself/herself is blue-eye or not. But after the first day, if the other one did not leave, he/she can confirm that he/she has blue eyes. Then they will leave together. If *x*=*k* (*k*>=2 and *k*<=n), for each person with blus eyes, they know that the number of people with blue eyes should be *k*-1 or *k*. To choose an arbitrary blue-eye person *A*, from his/her perspective, if other *k*-1 people did not leave at first *k*-1 day, then he/she can confirm that he/she is blue-eye and they can leave together at the *k* day.

Actually, this solution is tricky because we see from the God view.

6.7 See the solution with codes at Solution07.java.

6.8 Since it is going to minimize the number of drops for the worst case, we should define what is the worst case. The worst case is that after egg one breaks, we still have many steps to test. Therefore, let egg one drops in this way: first 10^{th} floor, and then 20^{th} floor… finally 100^{th} floor. If it breaks, then *N* must be in the range of [k*10,(k+1)*10]. After that, we let egg two drops in this way: first (k*10+2)th floor, and (k*10+4)th floor… and finally (k*10+10)th floor.

6.9 See the solution with codes at Solution09.java.

6.10 For this problem, I firstly only generate the naive solution (28days). After reading the textbook, I find it really impressive to use binary representation. Since we have 10 strips, we can handle at most $2^{10} = 1024$ bottles of soda. Each strip tests one bit, and thus we can get the exact poisoned bottle. For example, strip 1 is responsible to test all the bottles whose last bit is 1. If it is positive, the number of poisoned bottle in this bit must be 1. Therefore we can find the poisoned bottle in 7 days.