12^{th} TuiJinBan, Quiz 4
Group B: Jiayang Gao *2019.07.05*

I give up all sell-side questions (4-7).

1. Given a fifty-seat room where each seat is numbered, people entered the room in ordered. The first one is drunk and he will take a seat randomly. The rest people will take their own seat as long as it is not taken. Calculate the probability that last two people can take their own seats.

   Answer. • I initially misunderstood the problem to be "the probability that **at least** two people can take their own seats". I keep my answer to that question any way. My answer to the original problem can be found under the next bullet point.

   The answer is 1, subtract by the probability that no one gets his own seat, and the probability that at least 1 gets his own seat.

   If no one gets his own seat, it has to be the case that person i take the seat of person $i+1$. So the probability is $1/50!$.

   If person k ($k = 2, \dots, 49$) takes his own seat and every one else takes the wrong seat, it has to be the case that person $k-1$ takes the seat of person $k+1$, and every other person i takes the seat of $i+1$. The probability is $(51-k)/50!$.

   Thus the probability is

$$1 - \frac{1}{50!} - \sum_{k=2}^{49} \frac{51-k}{50!} = 1 - \frac{1+\ldots+49}{50!} = 1 - \frac{1}{2*48!}$$

   - This is equivalent to seat 1 is taken before either 49 or 50 is taken. Whenever a person makes a random choice, he has same probability of taking $1, 49, 50$ (if they are all available). So the probability that 1 is taken first among the three is $1/3$.

   □

2. There are 5 girls who need to guard a store from Mon-Fri. Calculate the number of possible arrangements given the constraint: (a) Everyone works twice a week. (b) Two girls guard the store in a single day and no two girls can work together twice.

   Answer. First, splits the girls into 5 groups of 2 each, satisfying both conditions. Then there are $5! = 120$ ways to assign those 5 groups into the 5 days.

   For the groups, let the girls be nodes and connect each pair of girls in the same group with edges. Thus, each node has 2 edges. Moreover, since no 2 nodes can be connected twice, the only way is to have a cycle with 5 nodes. There are 4! possibility of such cycle with directions, and thus $4!/2 = 12$ possibilities of such cycle without direction.

   Thus the answer is $120 * 12 = 1440$. □

3. Follow up, what if there are 7 girls and 7 days schedule to be set up, what is the number of possible arrangement?

   Answer. Same argument as above. There are $7! = 5040$. possibilities to assign 7 groups into 7 days.

   And for the groups, there're now two kinds of cycles: (1) a cycle with 7 nodes, (2) two cycles with 3 and 4 nodes each. So the total possibility is

$$\frac{6!}{2} + \binom{7}{3}(\frac{2!}{2} * \frac{3!}{2}) = 465.$$

   So a total of $5040 * 465 = 2343600$ possibilities.

   □

4.

5.

6.

7.

8. Consider the basketball shooting game, define success rate as number of successful shoots divided by number of total shoots. Assume the successful rate rising from below 0.5 to above 0.5, is there a moment which has exactly success rate 0.5.

   Answer. Yes. Otherwise, there exists a point, such that in the previous round success rate < 0.5 but in the current round success rate > 0.5. This is equivalent to having a, x integers such that

$$\frac{a}{2a + x} < 0.5 \implies x \geq 1$$
$$\frac{a + 1}{2a + x + 1} > 0.5 \implies x < 1.$$

   This leads to a contradiction.

   □

9. Given two strategies A and B, as well as the corresponding P&L of these strategies on each day. If one is going to be shut down, how to decide which one to shut?

   Answer. • Keep the one with higher Sharpe ratio, which is

$$\sqrt{252}\frac{mean(r_t)}{sd(r_t)}$$

- If sharpe similar, keep the one with less correlation with other strategies in the book.
- Also if the Sharpe ratio is calculated using a long history, we should also note recent year performance. It's very possible that a strategy that used to perform well in old years has decayed in the recent years. This is not immediately reflected in Sharpe ratio calculation.

□

10. Design an algorithm to check is a point is inside the triangle or not.

    Answer. Answer a more general question: check whether a point is inside a convex polygon. (Or the same method can be used to answer whether a given set of nodes and edges form a convex polygon.)

    This is equivalent to, given the line defined by any two neighboring points, the given point is on the side of the line with all other points in the polygon. Because the polygon is convex, we know that all nodes in that polygon lie on the same side of the line. Thus we only need to check whether the given point is on the same side of the line as any other point in the polygon.

    Given point X, Y, and line defined by A, B, we know that the cosine between $X - A$ and $B - A$ has the same sign as the dot product $(X - A) \cdot (B - A)$. Also that X, Y is on the same side of the line $A- > B$ if and only if $(X - A), (B - A)$ and $(Y - A), (B - A)$ have cosines of the same sign. Thus we only need to check whether $(X - A) \cdot (B - A)$ and $(Y - A) \cdot (B - A)$ have the same sign.

    The complexity is linear in number of nodes in the polygon.

    □

11. Design an algorithm to check palindrome sequence.

    Answer.

```
for i in range(0, int(len(S)/2)):
    if $S[i]! = S[-(i+1)]$:
        return False
return True
```

□

12. Given an array, design an algorithm to return the longest decreasing sub-array.

Answer. Loop through the array once, and keep the index of the start s and end e of the current longest decreasing sub-array, as well as the index c of the start of the current decreasing sub-array.

At each index $i-1$, compare if $S[i] < S[i-1]$. If so, the current decreasing sub-array continues and do nothing. If not (or if $S[i]$ does not exist), the current decreasing sub-array ends at $i-1$. Compare the length of the current sub-array $i-c$ and $e-s+1$. Update s and e to be c and $i-1$, if the former is longer. Then set $c=i$ and move on if $S[i]$ exists.

Finally return $S[s : e+1]$.

□

13. Implement the Sudoko algorithm.

Answer. The idea is to use back-tracking algorithm. Order each empty cells. And try to fill in each empty cell in order with possible values in the order of 1 to 9. After each try, check whether the corresponding row, column, and box has duplicate. If so, move on to try the next number.

When we reach a point where no number can be filled in the current empty cell, we back-track and go back to the last filled cell. Try to replace the current number with a larger number. If this is not possible, we back-track again. □

14. Consider a chess board with each cell has a value on it, you can only walk right or down, find the path from up-left to down-right which has largest value.

Answer. Use dynamic programming. Let $(0,0)$ be the down-right point, and let $V(i,j)$ be the total value from (i,j) to $(0,0)$ under optimal path, and let $P(i,j)$ be the corresponding path. Let $x(i,j)$ be the value value at (i,j).

Then we have

$$\begin{aligned}
V(i,j) &= x(i,j) + \max(V(i-1,j), V(i,j-1)) \qquad \text{if } i>0, j>0\\
V(i,0) &= \sum_{k=0}^{i} x(k,0)\\
V(0,j) &= \sum_{k=0}^{i} x(0,k).
\end{aligned}$$

The corresponding $P(i,j)$ can be updated accordingly (by inserting the maximizer index to the beginning of the path).

The path that we are looking for is simply P at the up-left point.

□