Quiz 4

Group 2

1. Problem 1

Let p_n denotes the probability that the last two persons will take their own seats. If $n = 3$, it is easy to see that $p_3 = 1/3$. In general, we also have $p_n = 1/3$. This can be proved by induction. In fact, suppose that it holds up to n. Then for $n+1$, if the first person takes his own seat (with probability $1/(n+1)$), then the last two persons will definitely take their own seat. If the first person takes one of the seats of the last two persons (with probability $2/(n+1)$), then the last two persons will NOT take their own seat. If the first the person takes the seat i, for $1 < i < n$ (with probability $1/(n+1)$), then the person i will essentially becomes the first person and the probability the last two persons will take their own seat is p_{n+1-i}, and by induction hyposthesis they are all $1/3$. So $p_{n+1} = \frac{1}{n+1} + \frac{1}{n+1}\sum_{i=2}^{n-1} p_{n+2-i} = \frac{1}{n+1} + \frac{1}{n+1}\frac{n-2}{3} = \frac{n+1}{3(n+1)} = 1/3$.

2. Problem 2

(a) Let A_i be the event that at day i there is no girl to guide the store. Then total way $= \binom{5}{2}^5 - |A_1 \cap \cdots \cap A_5|$. But by inclusion exclusion principal we know $|A_1 \cap \cdots \cap A_5| = \sum_{i=1}^{5} |A_i| - \sum_{1 \le i < j \le 5} |A_i \cap A_j| + \cdots + |A_1 \cap \cdots \cap A_5|$. But $A_i = \binom{4}{2}^5$, $|A_i \cap A_j| = \binom{3}{2}^5$ for $i \neq j$, $|A_i \cap A_j \cap A_k| = 1$ for $i \neq j \neq k$ and $|A_{i_1} \cap \cdots \cap A_{i_4}| = |A_1 \cap \cdots \cap A_5| = 0$. So $|A_1 \cap \cdots \cap A_5| = 5 * \binom{4}{2}^5 - 10 * \binom{3}{2}^5 + 10 * 1$. Hence the total ways is $\binom{5}{2}^5 - (5 * \binom{4}{2}^5 - 10 * \binom{3}{2}^5 + 10 * 1) = 100000 - 38880 + 2430 - 10 = 63540$.

(b) There are $\binom{5}{2} = 10$ different pairs of girls. Total ways are $\binom{10}{5} * 5! = 252 * 120 = 30240$.

(c) If both under the constraints of (a) and (b) then the answer is as follows: first we count the number of ways to form five pairs. girl 1 has $\binom{4}{2} = 6$ different ways to choose two other girls to form a pair, and for each two choices, the rest has two different choices to form the rest pairs: for example, if girl 1 chooses girls 2 and 3 to form two pairs, then girls has to pick her the other partners from girls 4 and 5 (girl 2 can not pick girl 3 since in that case there is only one choice for girl 4 (or 5) to pick her partners). So the number of ways to form five pairs is $6 * 2 = 12$. For each five pairs there are 5! different ways to guide the store. So the total number of arrangement is $12 * 5! = 1440$.

3. Problem 3

similar to 2

4. Problem 4

compute direct.

5. Problem 5

W_t^n is a martingle iff $N = 1$.

6. Problem 6

Problem of green book: Let $V = \frac{1}{S_t}$. Applying the Ito's differential to V gives

$$dV = (-r + \sigma^2)V dt - \sigma V dW_t.$$

So V follows a geometric distribution At last we get $V = \frac{1}{S_t} e^{-2r(T-t)+\sigma^2(T-t)}$.

7. Problem 7

no idea

8. Problem 8

yes

9. Problem 10

welch test and sharpe ratio

10. Problem 10

```
double area(double x1, double y1, double x2, double y2,
                double x3, double y3) {
    return abs((x1*(y2-y3)+x2*(y3-y1)+x3*(y1-y2)));
}

bool isInside(double x1, double y1, double x2, double y2,
              double x3, double y3, double x, double y) {
  double total = area(x1,y1,x2,y2,x3,y3);

  double A1= area(x,y,x2,y2,x3,y3);

  double A2= area(x1,y1,x,y,x3,y3);

  double A3= area(x1,y1,x2,y2,x,y);

  return (total == A1+A2+A3);

}
```

11. Problem 11

```
string reverse(string &str) {
    int n = str.length();
    if (n < 2) return str;
```

```
    int i = 0;
    int j = n-1;
    while (i < j) {
        swap(str[i],str[j]);
        i++;
        j++;
    }
}


bool checkPalingdrome(string str) {
    string str1 = str;
    reverse(str);
    for (int i =0; i < str.length(); ++i) {
        if (str1[i] != str[i]) return 0;
    }
    return 1;
}
```

12. Problem 12

dynamic programming. There is a O(NlogN) algorithm using binary search + DP. https://swiyu.wordpress.com/2012/10/15/longest-increasing-subarray/

13. Problem 13

back tracking

14. Problem 14

dynamic programming.