

Qishi Quiz 3 Solution by Group C

1 Math/Stat.

P5

covariant matrix is symmetric positive semidefinite. According to Sylvester's criterion, its principal minors are greater than or equal to 0.

$$\Omega = \begin{vmatrix} 1 & r & 0 \\ r & 1 & r \\ 0 & r & 1 \end{vmatrix}, |\Omega| = 1 - r^2 > 0, \begin{vmatrix} 1 & r \\ r & 1 \end{vmatrix} > 0 \implies -1 < r < 1$$

P6

S_n as a symmetric random walk. (Where $S_n = X_1 + X_2 + X_3 \cdots + X_n$ with n being the stopping time). We know that both S_n and $S_n^2 - n$ are martingales.

a) Let $P(-1)$ be the probability that the drunk man go through the left door, and N be the stopping time. $E[S_N] = P(-1) * (-1) + (1 - P(-1)) * 99 = S_0 = 0$, $E[S_N^2 - N] = E[P(-1) \times 1 + (1 - P(-1)) \times 99^2] - E[N] = S_0^2 - 0 = 0$, $\implies P(-1) = 99/100, N = 99$

b) if left door is locked, the drunk man need to walk right for the next step at the left door, and exit from the right door. we could use symmetric analysis, it's equivalent that the drunk man could exit at either 99 or -101. Use the same method, we get $P(99) = 0.505, N = 9999$

2 Programming.

P16

We flip a coin n times to generate a binary sequence of size n . This number is in the range of $0 - (2^n - 1)$. If we treat binary sequence $000 \cdots 0$ as value 2^n , we get a variable ranging from $1 - 2^n$. This number mode 6 will generate output $0 - 5$ equally if we reject number $2^n > 6^m$ given we want m die flips simultaneously. $m = \lceil n \log_3 2 \rceil, r = 1 - \frac{3^m}{2^n}$, r is the rejection rate.

P17

Basically, we move both robots in the same direction, the robot which gets to zero first stays at zero, the other robot reverts its direction until it gets to zero as well. Two robots will meet at zero.

```
#include <iostream>
using namespace std;
```

```
class robot{
private:
    int pos;
public:
    robot(int x){
        pos = x;
    }
    void Go_Left(){
        pos -= 1;
    }
}
```

```

    void Go_Right(){
        pos +=1;
    }
    bool at_zero(){
        return ( pos == 0 );
    }
};

int main(){
    robot robot1(-8);
    robot robot2(7);
    int count=0;
    while( !robot1.at_zero() || !robot2.at_zero() ){
        if ( robot1.at_zero() || robot2.at_zero() ){
            if( robot1.at_zero() ) {
                robot2.Go_Left();
                count++;
            } else {
                robot1.Go_Left();
                count++;
            }
        } else {
            robot1.Go_Right();
            robot2.Go_Right();
            count++;
        }
    }
    cout << count << endl;
    return 0;
}

```