JIE MEI z5173405

Question 3

According to the question, since each day has three different possible activities in n days and two adjacent days cannot be the same activity, we can use a two-dimensional array Activity[N][3] to store the enjoyment value of each activity every day. Activity[i][j] represent the value of enjoyment doing jth activity on day i (j = 0,1,2). Then we also need to create new 2-dimensional array res[N][3] that represent doing jth activities within i days to get the maximum enjoyment value. Traverse the entire array to calculate the maximum enjoyment value of doing jth activities every day. Finally, we can get the 3 maximum enjoyment value: res[N-1][0], res[N-1][1], res[N-1][2]. The maximum value of these three values is the final answer.

Sample code(java)

```java
class Q3 {
    public static int[][] maxEnjoyment(int rows,int columns,int[][] activity) {
        int[][] res = new int[rows][columns];
        res[0][0] = activity[0][0];
        res[0][1] = activity[0][1];
        res[0][2] = activity[0][2];
        for(int i = 1; i < rows; i++) {
            for(int j = 0; j < 3; j++) {
                if(j == 0) res[i][j] = activity[i][j] + Math.max(res[i-1][1], res[i-1][2]);
                else if(j == 1) res[i][j] = activity[i][j] + Math.max(res[i-1][0], res[i-1][2]);
                else if (j == 2) res[i][j] = activity[i][j] + Math.max(res[i-1][0], res[i-1][1]);
            }
        }
        return res;
    }

    public static void main(String[] args) {
        int rows = 3;
        int columns = 3;
        int[][] activity = new int[rows][columns];
        for(int i = 0; i < 3; i++) {
            for(int j = 0; j < 3; j++) {
                activity[i][j] = 2;
            }
        }
        activity[0][0] = 3;
        activity[0][2] = 3;
        activity[1][1] = 4;
        activity[1][2] = 5;
        int res[][] = new int[rows][columns];
        res = maxEnjoyment(rows, columns, activity);
        for(int i = 0; i < 3; i++) {
```

```
                for(int j = 0; j < 3; j++) {
                    System.out.print(res[i][j]);
                }
                System.out.println();
            }
            System.out.println(Math.max(Math.max(res[2][0], res[2][1]),res[2][2]));
        }

}
```

The example activity[3][3] like that:

3	2	3
2	4	5
2	2	2

The res[3][3] should be that:

3	2	3
5	7	8
10	10	9

The code output:

```
80000gn/T/vscodesws_6b1b6/jdt_ws/jdt.ls-java-project/bin Q3
323
578
10109
10
bash-3.2$
```