E. Okabe and El Psy Kongroo(Codeforces 821)

`DP` `Matrix_quick_power`

从非负整点网格中的(0,0)走到(k,0)，每一步只能走到右上、正右和右下三个点，要求不能超过给定的n条水平线(a1,b1,c1),(a2,b2,c2),...,(an,bn,cn)。其中，ci为纵坐标且横坐标不间断。求所有可能路线的数量。

链接

[Codeforces 821-E]

分析

1. 动态规划DP
   设状态d(i,j)表示从(0,0)到(i,j)的所有路线的数量。其中，0<=i<=k,0<=j<=max{ci}。初值为0。状态转移方程为

   ```
   d(i,j) = valid(i,j)==1 ? (d(i-1,j-1)+d(i-1,j)+d(i-1,j+1)) : 0;
   ```

   (边界另行处理)。其中，valid(i,j)表示(i,j)是否可以走。
   边界条件：d(0,0)=1
   最终解：d(k,0)
   由于k=$O(10^{18})$，不仅时间复杂度会爆炸，内存也会超。
   认真观察状态转移发现，i其实没用，只是机械地加1，可以去掉。于是

$$\begin{bmatrix} d(15) \\ d(14) \\ d(13) \\ \vdots \\ d(2) \\ d(1) \\ d(0) \end{bmatrix}$$

   表示当前列各点的路径数，则状态转移方程为

$$\begin{bmatrix} dn(15) \\ dn(14) \\ dn(13) \\ \vdots \\ dn(2) \\ dn(1) \\ dn(0) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} d(15) \\ d(14) \\ d(13) \\ \vdots \\ d(2) \\ d(1) \\ d(0) \end{bmatrix}$$

   只需保存d(15),d(14),...,d(0)这16个状态。这就解决了空间爆炸的问题。
   为解决时间爆炸的问题，使用矩阵快速幂，类似于[求n很大时的斐波拉契数]。

2. 矩阵快速幂
   矩阵快速幂可以在*O(log n)*时间里求A^n。
   核心代码：

```
while(n){
       if(n & 1) Matrix(cot,temp);   //如果是奇数
       Matrix(temp,temp);
       n /= 2;  //不断除2
}
```

在这里，

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 \end{bmatrix}$$

由于n很小，分段求即可解决有效值的问题。