【算法分析】

首先考虑没有石墩的情况，设n为石墩数，m为荷叶数，设F[n][m]表示当有n个石墩，m个荷叶时，能跳过去的最多妖兽数。

显然F[0][0]＝1，F[0][1]＝2，F[0][2]＝3，……可以归纳出F[0][y]＝y＋1。即先把y只妖兽放在每片荷叶上，让最后一只直接跳过去，然后荷叶上的妖兽依次跳过去。

在F[n][m]的基础上增加一个石墩，此时就有n＋1个石墩了，把第n＋1个石墩看成右岸，这样就可以把F[n][m]个妖兽跳到第n＋1个石墩上（借助原来河里的n个石墩，m个荷叶），现在河里还有n个空石墩，m片空荷叶，还可以帮助F[n][m]个妖兽从左岸跳到真正的右岸，把第n＋1个石墩看成左岸，借助河里的n个石墩，m片荷叶，顺利地跳到右岸妖兽身上。至此一共可以跳过去2×F[n][m]只妖兽。

由此可知：关系式F[n＋1][m]＝2×F[n][m]

推导：F[n][m]＝2×F[n－1][m]

＝4×F[n－2][m]

……

＝(2i)×F[n－i][m]

……

＝(2n)×F[0][m]

当n＝0时，河里只有m个荷叶，每片叶上只能有一只妖兽，再加上从左岸可以直接跳到右岸的一只，所以共有m＋1个妖兽，即F[0][m]＝m＋1，所以F[n][m]＝(m＋1)×2n。

|  |  |
| --- | --- |
| man | 其实此题就是汉诺塔的改进版，把荷叶看成一个整体，不要被题目的形式所迷惑。 |

参考程序如下所示。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | //妖兽特攻队  #include <bits/stdc++.h>  using namespace std;  int main()  {  int m,n;  scanf("%d%d",&n,&m);  m++;  for(int i=1;i<=n;i++)  m\*=2;  printf("%d",m);  return 0;  } |