Solution

清华大学

高天宇

Box

- 题目大意：
  - 有n个盒子，每个盒子有一个载重量。将这n个盒子摞成若干列，问最少多少列。
- 部分分：
  - 搜索
- 正解：
  - 先考虑这样一个问题，你现在有若干列盒子。现在有一个新的盒子，你只能挑一列，把新盒子放到最底下。为了后续策略最优，你应该怎么放？

Box

- 显然是应该找（能放的）个数最多的那个放
- 因为这样放，对于后续的操作来说才是最优的。
- 比如原来是2，3，5，再放肯定要放在5的底下，变成2，3，6。对于后续的决策来说，2，3，6肯定比3，3，5或者2，4，5优。
- 所以我们就有了一种贪心的方法。先将所有的盒子按照承载量从小到大排序。

Box

- 然后我们开一个数组，记录一下当前一共有多少列，每一列一共有多少个盒子。从小到大扫描所有的盒子，找到**能放下的**数量最多的列，放进去。如果没有任何一列能放下，则建一个新列。
- 如目前：2，4，6。新来了一个承载量5的盒子，就应该放在『4』那一列。
- 举例：0，1，1，2，2，3

Castle

- 题目大意：有n个点，每个点有一条出边。要求前k个点能走到1号点，后k个点不能走到一号点，问方案数。

Castle

- n<=8
  - 直接爆搜每个点的出边指向谁，然后检查即可。
- n<=10^5
  - 我们发现，前k个点肯定和前k个点互相连边。后n-k个点肯定不会连到前k个点里面去。
  - 所以，我们只要爆搜前k个点连接的方案，然后检查；后n-k个点，只要连的是后n-k个点，爱怎么连怎么连，方案数是(n-k)^(n-k)。最后把两部分方案数乘起来就行。
- n<=10^18
  - 快速幂即可。

Jump

- 最简单的DP比较好想
- 令f[i][j]为从i这个点开始，下一步走j个，能够收集到的最多的宝藏
- f[i][j] = a[i] + max(f[i+j][j-1],f[i+j][j+1],f[i+j][j])

Jump

- n^2复杂度，受不了
- 如果移动步数可以固定就好了
- 移动步数会发生偏移
  - 最多偏移多少呢？
- 初始步数为1，每次多走1步，走n次
  - n(n+1)/2
- 偏移量最多是sqrt(2n)量级的
- 枚举时只需要枚举初始步数+-最大偏移量即可