1

比 k 小可以分成若干类，枚举第一位不同的地方，接下来就要求是某个数的子集。

然后能选就选。

复杂度$O(n*30)$

2

dp，限制相当于子序列中每个数选一个是 1 的位 i，下一个数也选一个是 1 的位 j，$i=j$ 则可以接上。因此$f(i,bit)$ 表示当前第 bit 位为1的最大dp值。

复杂度$O(n*30)$

3

考虑dp，可以先考虑求方案数而不是权值和以方便思考，实际上两者转移是类似的。

$f(i,l,r)$ 表示只考虑第 $[l,r]$ 这些行，第 i 列以后的这部分的方案数。

对于一个固定的 i,l,r ，显然可以再dp辅助计算，$dp(k,c)$ 表示考虑完前 k 行，当前这位最大填了 c 。每次转移加一段 $c+1$，利用 $f(i+1)$ 的信息来转移。

这是最暴力的办法，可以想得比较清楚，复杂度$O(10mn^4)$。

考虑优化，很明显对于每个 l，不同的 r 的内层 dp 几乎都是一样的，放一起做就好了。本质上就是内层dp只需要先枚举 i,l 即可计算。复杂度$O(10mn^3)$。

注意转移的复杂度别写残，比如多乘个 10 啥的。

权值和也一样，再设个 $g(i,l,r)$ ，转移也差不多，很容易自己脑补出来或者可以看标程。

4

首先分裂这个过程不方便思考，我们考虑每个石子而不是每堆石子。一次操作相当于给每个石子分配01标号，满足每次标1的石子不超过m个。这样每个石子对应一个等长的01串，我们的目标是01串两两不同。

二分答案，考虑如何判定某个mid是否合法。一个必要条件是$mid\cdot m$足够大，即存在一组方案使得01串两两不同且1的个数$\le mid\cdot m$。事实上这也是充分的，我们给出构造来说明这一点。首先一个贪心是按1的个数从小到大放，显然1的个数不是最大的那些串的1是平均分配的，问题转化为要取若干个$[1,n]$的m元子集，使得每个数出现次数极差≤ 1。我们随便构一组解，如果不合法就找出一个出现次数太大的位i，一个出现次数太小的位j，显然存在一个01串s满足$s[i]=1,s[j]=0$，且不存在另一个01串t满足$t\oplus s=2^i\oplus 2^j$。因此我们只需$swap(s[i],s[j])$。迭代显然可以有限次结束。

具体实现需要求组合数，这里不能取模但是由于$n\le 10^9$所以直接写不会爆`long long`。组合数的增长是很快的因此暴力的复杂度可以接受。