1000 pt

这道题目其实挺难的。主要麻烦在于找到一种合适的枚举方法，能够将所有可能的情况尽量以与11相关的复杂度而不是与n相关的复杂度。

最后是实现了一下题解的方法：

题解的方法主要分为4步：

1. 增加限制条件，如果每个数字都需要用上。将0~9放在一个集合里，{0 ,1,2,3,4,5,6,7,8,9}，然后枚举出这个集合的所有划分。在每个划分下的每个子集的所有数字就能够组成一些合理的数字，这些情况乘起来就是一个划分下的数字组成个数。
2. 去掉限制条件，也就是允许有一些数字不使用。考虑每个划分中，随便选定一个集合（或不选），将其当做不使用的数字，这样在每个划分中就包括了有些数字不使用的所有情况。
3. 进一步简化，引入一个-1，将其加入到数字中。还是回到限制条件，这样每个划分必须有一个子集包含-1，这个子集就当做不使用的数字忽略掉。于是问题就规约到了求-1~9组成的集合的所有划分的问题上了。
4. 由于总共只有11个数字，则真子集的可能情况只有2^11-1种，于是每种情况下的合理数字个数可以预计算。只需要把真子集中的数字由小到大所有合法排列加起来即可。
5. 求所有划分，使用递归回溯的方法：
6. 用一个数组par[i]记录以i为最小元素的子集中的元素。
7. F(a , p)，p为当前考虑到第p个元素，a为当前已经被考虑过的元素的情况。
8. 对第p个元素，如果其已经被前面的子集包含掉，则直接返回f(a , p+1)

如果没有被包含掉，就枚举所有以p为最小元素的真子集的所有可能，计算f(a’,p+1)并累加。

1. 对最后一个考虑的真子集，也就是9为最小元素的真子集时。对当前的划分进行计算。

我一直犯的错误就是认为这个递归回溯可以memo。但实际上是不行的：因为计算划分时是根据当前的par[0~10]计算的，对f(a,p)而言，前面确定了的par[0,p-1]是会影响其结果的。所以这个回溯必须是指数复杂度的，也就是O(11\*2^11\*2^(11\*11))。第一项是p的可能，第二项是a的可能，第三项是par[0~10]的可能。