nCrを分ける

以下のpythonの文で生成される数列をTとする。
また、nは自然数、関数Cは対応する二項係数を返すものとする。

```
T = []
for i in range(0,n + 1):
  T.append(C(n,i))
```

　Tの各要素をk個のグループに分け、それぞれのグループの和の集合をKとする。
この時、max(K) - min(K)が最小となるような分け方について考える。

- k = 1のとき
  グループが１つしかないため、 max(K) - min(K) = 0が常に達成できる。

- k = 2のとき
  nが奇数のときはTは左右対称になるため、対応する要素を互いに異なるグループに分けることでmax(K) - min(K) = 0が達成できる。nが偶数のときも、パスカルの三角形を考えるとT_(n - 1)の全ての要素が偶数番目の要素と奇数番目の要素に足されているため、同様にmax(K) - min(K) = 0となる。

- k = 3のとき
  要素の番号をiとして、3を法にとったときの値でグループ分けをする。

```
K = [0,0,0]
for i in range(0,n + 1):
  K[i % 3] += K(n,i)
```

　これが自明な最小値であるsum(K) - sum(K) = 1を達成できることを証明する。

パスカルの三角形を観察すると、

$$1_n = 1_{n-1} + 0_{n-1}, 2_n = 2_{n-1} + 1_{n-1}, 0_n = 2_{n-1} + 0_{n-1}$$

が成り立つことがわかる。n = 1のときの

$$(1_n, 2_n, 0_n) = (1, 1, 0)$$

という値と漸化式より、

$$(1_n, 2_n, 0_n) = (r, r, r-1) \, or \, (r, r-1, r-1)$$

という形を取ります。よってmax(K) - min(K) = 1が常に成り立つ。