**L.2,3 importについて**

itertoolsもtimeも標準ライブラリ

itertoolsはイテレータ（配列などのインデックス制御に用いる。）を生成するツール集

timeは処理時間計測のために仕様

**L.72 ----組み合わせの配列を格納した配列（2次元配列）を生成----**

(本番では28C14を想定して計算するが)

たとえば5C3なら

[[0,1,2],[0,1,3],[0,1,4],[0,2,3],[0,2,4],

[0,3,4],[1,2,3],[1,2,4],[1,3,4],[2,3,4]]

のように0から始まる組み合わせの配列を生成

**L.81 --コンビネーションに対応する着座表を作成--**

座席の位置(行・列)のみ格納した「小さな配置図」の中から、

組み合わせ配列にしたがって、「人が座る位置」だけ取り出した配列を作成

**L.87 --不快度の計算の下準備--**

いちいち大きな座席表を作って計算するわけにもいかないので、

各行・各列に着席している人数をそれぞれ

total\_people\_row, total\_people\_columnという配列に格納

その後のループは、

その行・列にいる人の数と、その人達の向いている方向により４通り場合分け

sum\_eyesightに視界の広さ×人数分

sum\_counted\_peopleに視界に入る人の数×人数分

を足していく

不快度は、視界に入る人の数の総和 / 視界の広さの総和 と定義する

すなわち、不快度 = sum\_counted\_people / sum\_eyesight

**※ 大きな座席図**

[[2. 0. 2. 2. 0. 2. 2. 0. 2. 2. 0.]

[0. 3. 0. 0. 3. 0. 0. 3. 0. 2. 3.]

[2. 0. 2. 2. 0. 2. 2. 0. 2. 2. 0.]

[2. 0. 2. 2. 0. 2. 2. 0. 2. 2. 0.]

[0. 3. 0. 0. 3. 0. 0. 3. 0. 2. 3.]

[2. 0. 2. 2. 0. 2. 2. 0. 2. 2. 0.]]

**※ 小さな座席図（座席がある場所の行・列）**

[[ 0. 1.]

[ 1. 0.]

[ 1. 2.]

[ 2. 1.]

[ 0. 4.]

[ 1. 3.]

[ 1. 5.]

[ 2. 4.]

[ 0. 7.]

[ 1. 6.]

[ 1. 8.]

[ 2. 7.]

[ 0. 10.]

[ 2. 10.]

[ 3. 1.]

[ 4. 0.]

[ 4. 2.]

[ 5. 1.]

[ 3. 4.]

[ 4. 3.]

[ 4. 5.]

[ 5. 4.]

[ 3. 7.]

[ 4. 6.]

[ 4. 8.]

[ 5. 7.]

[ 3. 10.]

[ 5. 10.]]

**※ 「人が座る位置」の配列**

上記の「小さな座席図」配列から、

各コンビネーションで指定されたインデックスに対応する組を取り出す

例：

コンビネーションが[0,1,4,6,17,24]と与えられた場合

小さな座席図 配列の[0,1,4,6,12,24]番目を取り出すので

[[ 0. 1.]

[ 1. 0.]

[ 0. 4.]

[ 1. 5.]

[ 5. 1.]

[ 4. 8.]]

以上が「人が座る位置」の配列である

**※ 実際のコード**では、

大きな座席図：ARRANGEMENT, min\_arrangement, max\_arrangement

小さな座席図：seat\_position

人が座る位置の配列：(ループ内)sat\_seat\_position, min\_seat\_position,

max\_seat\_position

コンビネーション配列：(ループ内)each\_combination, combination\_list[min\_index],

combination\_list[max\_index]

全コンビネーションを格納する配列：combination\_list