想法：n個挑k個排列組合

從1開始往後 起頭的i，往後增加都要比進去的大

If k =2

[1], [2]

[1,2->n] [2,3->2,n]

class Solution:

def combine(self, n: int, k: int) -> List[List[int]]:

visited = set()

res = []

z = -float('inf')

self.backtracking(res,visited,[],n,k,z)

return res

def backtracking(self,res,visited,subset,n,k,z):

if len(subset) == k:

res.append(subset)

for i in range(1,n+1):

if z < i :

visited.add(i)

z = i

self.backtracking(res,visited,subset+[i],n,k,z)

visited.remove(i)

class Solution {

public:

vector<vector<int>> combine(int n, int k) {

vector<vector<int>> result;

int i = 0;

vector<int> p(k, 0);

while (i >= 0) {

p[i]++;

if (p[i] > n) --i;

else if (i == k - 1) result.push\_back(p);

else {

++i;

p[i] = p[i - 1];

}

}

return result;

}

};