# 经验

## 一：接口类指针的参数

int\*\* subsets(int\* nums, int numsSize, int\* returnSize, int\*\* returnColumnSizes)

leetcode这种需要我们自己malloc出一个char\*\* res最后进行return。

对于returnColumnSizes 其实也是需要我们malloc出一个一维数组，然后赋值给\* returnColumnSizes。

res = (int\*\*)malloc(sizeof(int\*) \* (\*returnSize));

for (i = 0; i < \*returnSize; i++) {

res[i] = malloc(sizeof(int) \* (numsSize));

memset(res[i], 0, sizeof(int) \* numsSize);

}

res\_col = (int\*)malloc(sizeof(int) \* (\*returnSize));

\*returnColumnSizes = res\_col;

## 二：针对malloc出来的二维指针与二维数组的指针

二维数组的指针例如int vec[10][10],vec当作为参数传递时，会被降维度变成int\*类型。

而(int\*\*)malloc(sizeof(int\*)\*NUM)不管是否被当作参数传递，都不会被降维。

这样接口要看是怎么传递参数来实现。

// 由于是传递二维数组，将维了，所以访问要按一维的来

Void interface0(int\*\* vec, int col\_num){

// 访问第i行j列

\*((（int\*）vec) + I \* col\_num + j)

}

Void interface1(int\*\* vec, int col\_num)

{

// 访问第i行j列

Vec[i][j]

}

Int main()

{

Int I;

Int vec0[10][10] = {0};

Int\*\* vec1 = (int\*\*)malloc(sizeeof(int\*)\*10);

For (I = 0; I < 10; i++) {

Vec1[i] = (int\*)malloc(sizeof(int)\*10);

memset(vec1[i],0, sizeof(int)\*10);

}

}