

假设 array1 有足够的空间了，于是我们不需要额外构造一个数组，并且可以从后面不断地比较元素进行合并。

- 比较 array2 与 array1 中最后面的那个元素，把最大的插入第 $m+n$ 位
- 改变数组的索引，再次进行上面的比较，把最大的元素插入到 array1 中的第 $m+n-1$ 位。
- 循环一直到结束。**循环结束条件：当 index1 或 index2 有一个小于 0 时**，此时就可以结束循环了。如果 index2 小于 0，说明目的达到了。如果 index1 小于 0，就把 array2 中剩下的前面的元素都复制到 array1 中去就行。

功能代码：

输入一次 $m>n$ 的情况

输入一次 $m<n$ 的情况

特殊输入情况：

- 当 array1 为空，array2 不为空时，将 array2 的所有元素添加到 array1 中即可
- 当 array1 不为空，array2 为空时，就是上面的循环结束条件，直接返回 array1.
- 当 array1 跟 array2 都为空时，返回空。

```
• #include<iostream>
• using namespace std;
•
• int merge(int numa[],int m,int numb[],int n) {
•     int index1=m-1;
•     int index2=n-1;
•     while(index2>=0){
•         if(index1<0){
•             for(int i=0;i<=index2;i++){
•                 numa[i]=numb[i];
•             }
•             break;
•         }
•         if(numa[index1]>=numb[index2]){
•             numa[index1+index2+1]=numa[index1];
•             index1--;
•         }else{
•             numa[index1+index2+1]=numb[index2];
•             index2--;
•         }
•     }
• }
```

-
- `int main() {`
- `int numa[]={1,3,7,0,0,0};`
- `int numb[]={2,4,5};`
- `merge(numa,3,numb,3);`
- `for(int i=0;i<6;i++){`
- `cout<<numa[i]<<endl;`
- `}`
- `return 0;`
- `}`