1. 斐波那契额数列：

原本自顶向下递归, f(n) = f(n-1)+f(n-2), 可以采用迭代的方式自底向上开始, 采用递推的方式：这样返回a数组。

for(int i =2; i < a.length; i++){

a[i] = a[i-1] + a[i-2];

}

1. DP动态递推：
2. 一维结构，二维结构，三维结构。（取舍最优子结构）

路径计数：正常情况下，就是分治。(其实分治也是从上到下，再从下到上翻上来)

如果是反向推理，就是从最后一步开始推理。那么反向向左和向上推理。

状态转移方程, DP方程。



count[i,j] = count[i+1,j]+count[i,j+1] 由下到上依次累加。

1. 最长公共子序列问题：

反着从字符串尾进行思考，往前找，（分解子问题，和之前类似，分解到可以解决的子问题，然后从可解决的子问题开始递推，分治的反向思路（一般分治从复杂问题出发））