

分治

简单的将一个复杂问题切分为一组子问题(通过递归的方式)

Divide & Conquer + Optimal substructure 分治 + 最优子结构

- 1.缓存(状态的存储)
- 2.在每一步中淘汰次优的状态 只保留这一步最优状态

动归的终极形态是递推，直接转换为 for 循环这种自底向上的方式

```
•  $F[n] = F[n-1] + F[n-2]$   
  
•  $a[0] = 0, a[1] = 1;$   
  for (int i = 2; i <= n; ++i) {  
     $a[i] = a[i-1] + a[i-2];$   
  }  
  
•  $a[n]$   
  
• 0, 1, 1, 2, 3, 5, 8, 13,
```

动态规划 和 递归或者分治 没有根本上的区别 (关键看有无最优的子结构)

共性：找到重复子问题

差异性：最优子结构、中途可以淘汰次优解