

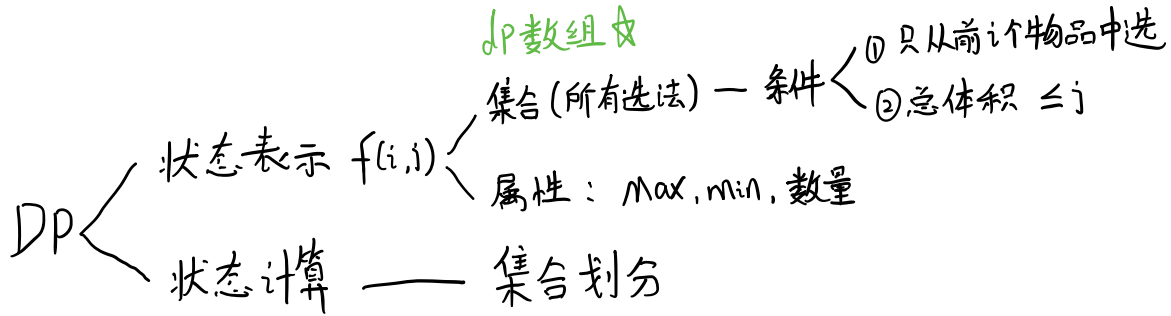
01背包 每件物品只用一次

完全背包 每件物品有无限个

多重背包

分组背包

## 一、背包问题



## 二、线性 DP

在一条线上进行 DP

什么是动态规划?

DP: 分阶段求解决策问题的数学思想  
不仅可用于编程领域, 也应用于管理学、经济学、生物学

复杂的问题分阶段进行简化, 逐步简化成简单的问题

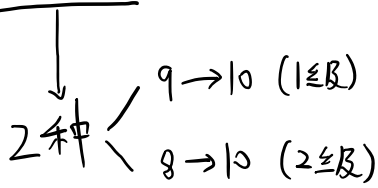
总结: 大事化小, 小事化了

例题: 有一座高度是 10 级台阶的楼梯, 从下往上走, 每跨一步只能向上 1 级或 2 级台阶  
求一共有多少种走法

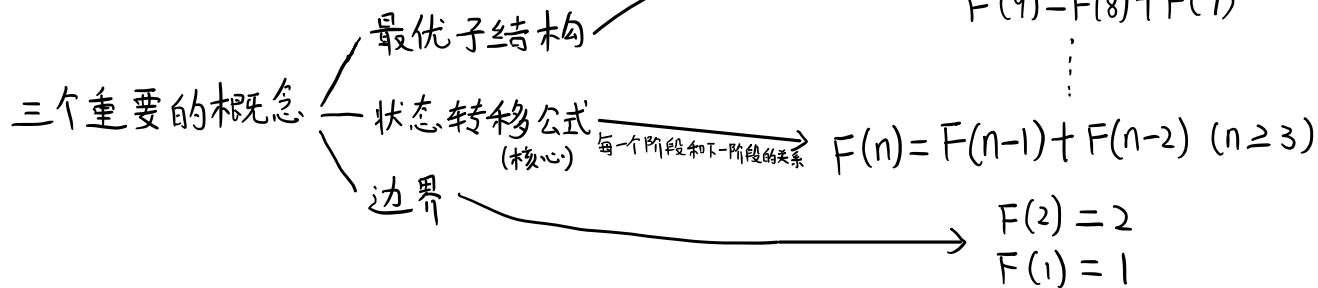
排列组合的思想, 写一个多层嵌套循环。每遍历一个组合让计数器加一 (暴力枚举)

问题建模:

① 假如只差最后一步就走到第 10 级台阶, 这时有几种情况?

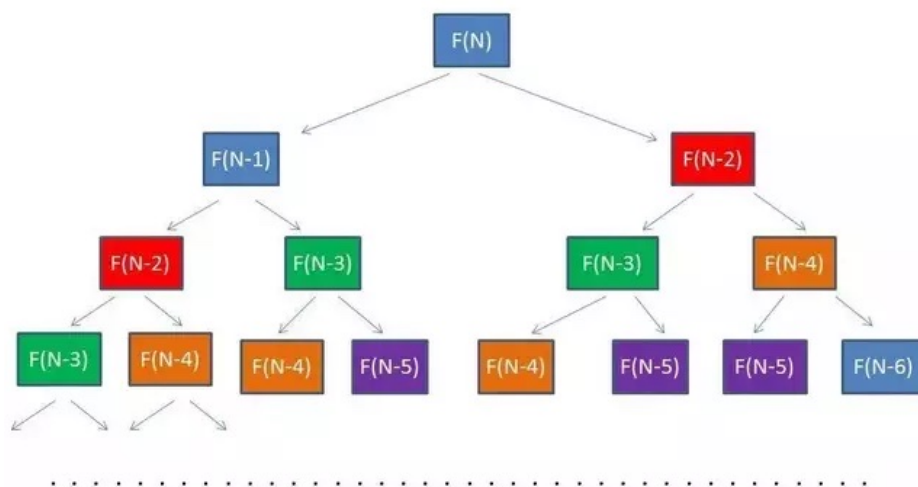


② 假如我们已知 0 到 9 级台阶的走法有  $x$  种  $\rightarrow$  0 到 10 级台阶的走法有  $x+y$  种  
 0 到 8 级台阶的走法有  $y$  种



求解问题是：

① 方法一：递归 时间复杂度  $O(2^n)$



② 方法二：备忘录算法 (集合map是一个备忘录) 时空复杂度为  $O(n)$

但这还不是真正的动态规划实现，我们不妨把思路逆转过来

③ 方法三：动态规划求解 自顶向下  $\rightarrow$  自底向上 时间  $O(n)$  空间  $O(1)$

每一次迭代的过程中，只要保留之前的两个状态就能推出新状态，不需方法二那样保留全部的子状态

科普: DP常用于有重叠子问题和最优子结构性质的问题  
并记录所有子问题的结果

两种解决问题的方式  $\left\{ \begin{array}{l} \text{自顶向下 (记忆化递归)} \\ \text{自底向上 (递推)} \end{array} \right.$