

# 1 introduction

动态规划是一种编程范式，可以系统地、有效地探索问题的所有可能解决方案。动态规划问题通常具有以下特征

1. 问题可以划分为可重叠的子问题，子问题的解可以被多次重用
2. 问题有最优子结构，即问题的最优解可以有子问题的最优解得出

动态规划（dp）的一个简单例子是斐波那契数列。对于第 $n^{th}$ 斐波那契数 $F(n)$ ，可以分解为两个小的子问题 $f(n-1)$ 和 $f(n-2)$ 。问题具有最优子结构，即 $f(n) = f(n-1) + f(n-2)$ 。子问题是重叠的，即对于 $f(5)$ 和 $f(6)$ 都需要计算 $F(4)$ 。

贪心问题与dp问题的区别在于，贪心问题有最优子结构，但没有重叠的子问题。分治算法与dp问题的区别在于，分支算法可以将问题分解为子问题，但子问题是不重叠的。

实现dp算法有两种方式，自下而上和自上而下自下而上的实现从最基本的例子开始迭代实现，在斐波那契数列的例子中，为了计算 $f(n)$ ，我们计算 $f(0), f(1), \dots$ 直到计算到 $f(n)$

```
1 F = array of length n+1
2 F[0] = 0
3 F[1] = 1
4 for i from 2 to n:
5     F[i] = F[i-1] + F[i-2]
```