

# 动态规划

# 简介

- 动态规划简称**DP**，是在**20世纪50年代**由一位卓越的美国数学家**Richcard Bellman**发明的。它作为一种重要的工具在应用数学中被广泛的应用。它不仅可以解决特定类型的优化问题，还可以作为一种通用的算法设计技术来使用。

# DP的实质

- 利用问题的所具有的重叠子问题的性质进行记忆化求解。（用空间换时间）

- 求Fibonacci数:

$$f(n) = f(n-1) + f(n-2) \quad n > 2$$

$$f(1)=f(2)=1$$

# 常规递归

```
int Non_DP(int n)
{
    if (n==1 || n==2)
        return 1;
    else
        return Non_DP(n-1) + Non_DP(n-2); □ }
```

- 指数级时间复杂度，无法忍受

# 两种实现方式

- 自底向上(bottom up)

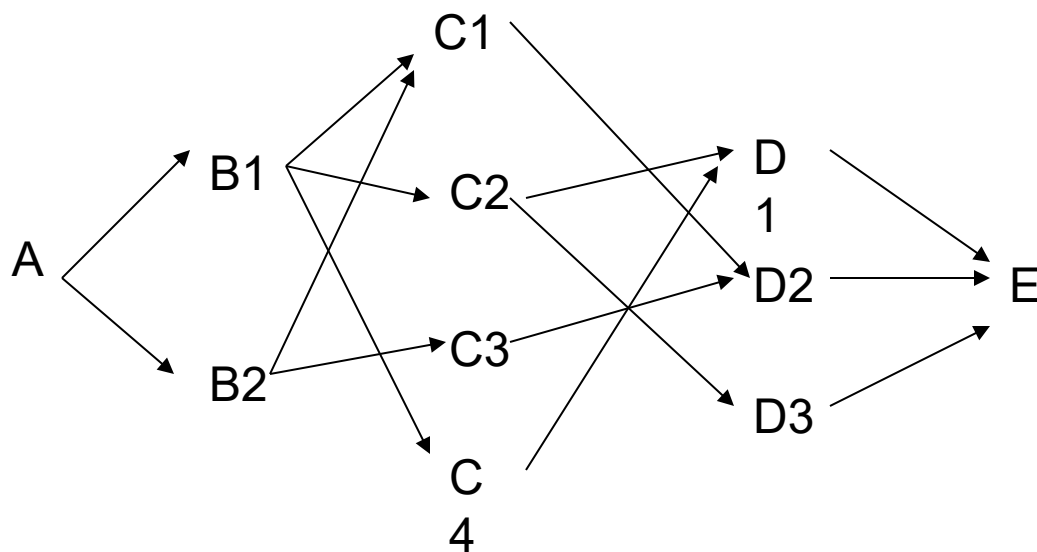
```
int DP_Bottom_Up(int n)
{
    memo[1] = memo[2] = 1;
    for (int i=3; i<=n; i++)
        memo[i] = memo[i-1] + memo[i-2];
    return memo[n]; □ }
```

- 自顶向下（top down）（这样写法也叫记忆搜索）

```
int DP_Top_Down(int n)
{
    if (n == 1 || n == 2) return 1;
    if (memo[n] != 0) return memo[n];
    memo[n] = DP_Top_Down(n-1) +
              DP_Top_Down(n-2);
    return memo[n]; □ }
```

# 基本概念

## ■ 最短路径问题



求A到E的最短路径。



- 直观的方法是用回溯法搜索。时间复杂度为指数级。
- 低效的原因：没有充分利用重叠子问题的性质。

- 此图有明显的次序，可以划分为5阶段。

