DP

#G面试准备

120. Triangle

Triangle - LeetCode

```
[
     [2],
    [3,4],
   [6,5,7],
  [4,1,8,3]
]
```

- 从顶向下
- 先初始化最顶点 以及三角形的两边 因为都只有一条路可走 中间的节点才是在两条路中选sum最小的一条
- 错: 初始化的时候

```
// opt[0][0] = 0;
opt[0][0] = triangle.get(0).get(0);
```

- 填完opt 最后在最后一行找最小值 输出

62. Unique Paths

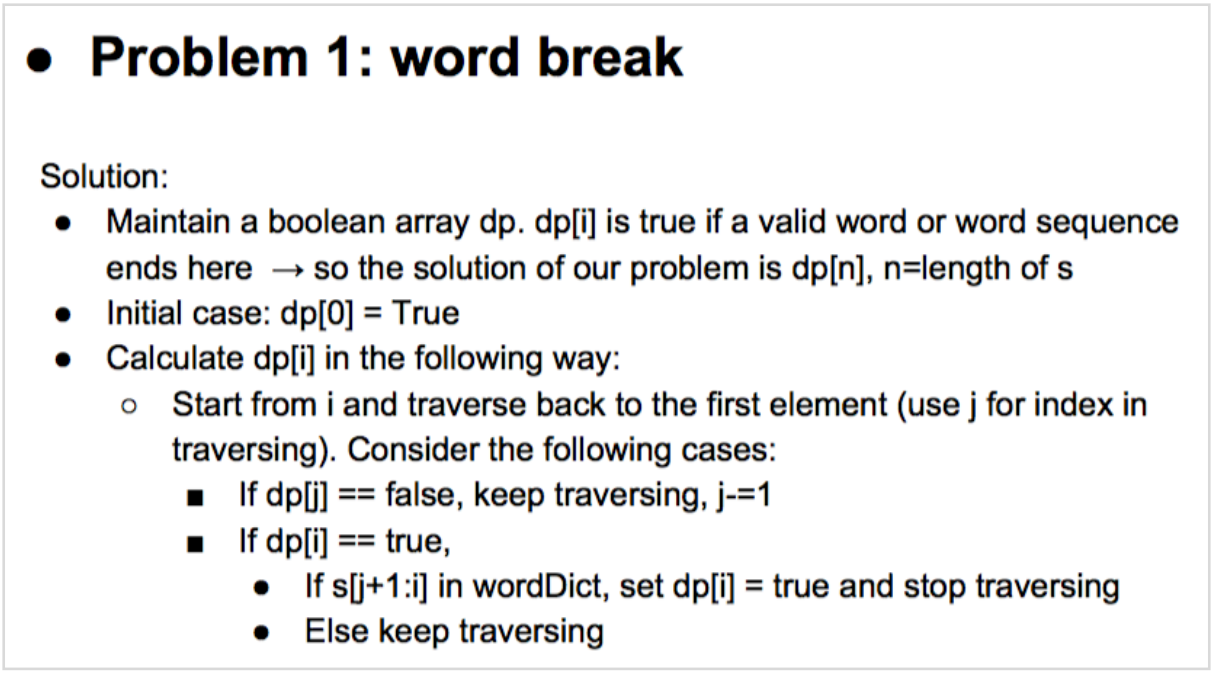
Unique Paths - LeetCode

- 数有多少种走法
- 初始化第0行和第0列都为1
  - 一开始写成一直累加1了 不对 只有一种走法
- 填剩下的格子 值为左边来的值加上上面来的值
- 返回最后一个格子的值等于多少

139. Word Break

Word Break - LeetCode

570复习课



- 做一个n+1位的opt boolean数组 第0位设置为true
- 对于找opt(i) 都要往前遍历到一个true的index j 然后看substring(j, i)在不在字典里
- 错: 只往i前面找了一个true 如果substring不在字典 不能直接返回false 要继续遍历j直到0 可能该substring包括进了之前被切的部分
- Follow up: **Word Break II** 见Backtracking/DFS

91. Decode Ways

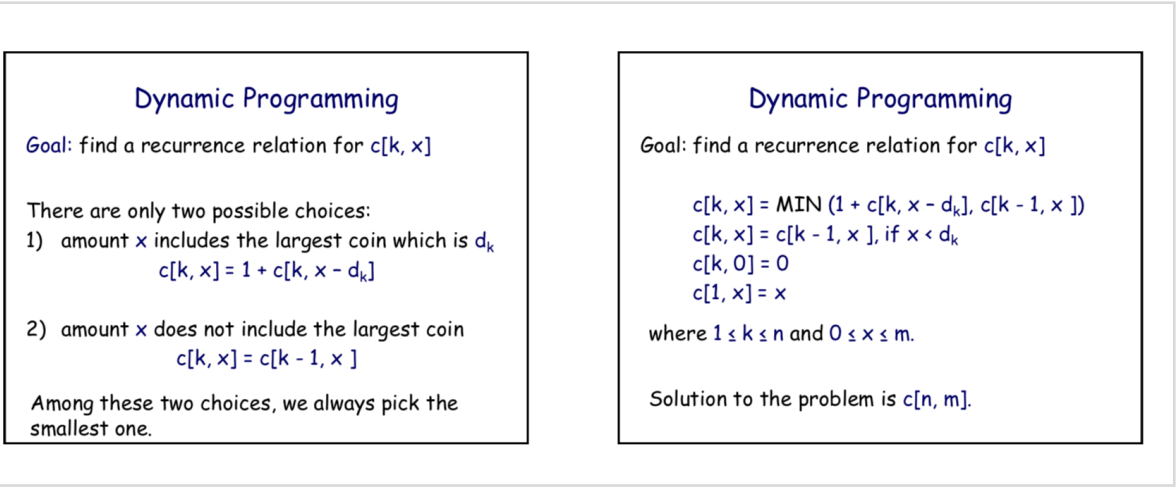
Decode Ways - LeetCode

- DP做法
- opt(i)表示从第0位到第i位这一串有多少种decode方法
- 还是要建一个n+1位的opt数组 例12321 这样对于第二位数就可以开始去看第opt(0)和opt(1)了
- 看一位: 当前位是否1-9 * opt(i - 1)
- 看两位: 前一位i-1和当前位i合在一起是否>=10 <= 26 * opt(i - 2)
- 错: 如果第0位是0 应该直接返回0 而opt(1) (第0位) 又初始化为1了 WA返回了1

322. Coin Change

Coin Change - LeetCode

- victor ppt: 二维opt数组



- opt(k, x)表示用前k种硬币表示数字x的最少方法
- k: 0 ➡ coins.length - 1, x: 0 ➡ amount, 总共amount+1个index
- 初始化: opt(0, x)用第0种硬币表示x
  - opt(k, 0): 都是0 表示数字0用0个硬币
- 初始化的错误: opt(0, x)
  - 如果能整除 就写硬币个数
  - 如果不能 要写为int max_value而不是0 要和0个硬币区别开
- 填二维数组opt(i, j) 第i种硬币 数字j 看两种情况
  - 不取硬币i opt(i, j) = opt(i - 1, j)
  - 取硬币i 需要满足以下条件
    - j - coins[i] >= 0
    - opt[i][j - coins[i]] != max_value 错这里 没判断他的上一个是否有答案 最后数字溢出
  - 取两种情况的最小值作为opt(i, j)
- 最后看opt[coins.length - 1][amount]是否还是max_value, 是就-1 不是就输出值
- 优化: opt只看上一行和同行前面的数 所以可以变成一维数组 两层for循环不断更新 最后取opt[amount]

动态规划和分治

> 1.动态规划是一种算法思想, 是高于算法的. 而分治的记忆化搜索是实现动态规划的一种手段.
> 2.那么什么是动态规划呢?
> -就感觉上来说, 动态规划的是"一层一层来", 基于前一个状态推出现在的状态.
> 3.动态规划为什么会快呢?
> -因为减少了很多不必要的重复计算.
> 4.动态规划和分治的区别?
> -动态规划约等于分治+记忆化, 因为有了记忆化, 所以算过的直接用就行, 就不用再算一遍了.
> 5.动态规划有方向性，不回头，不绕圈儿，无循环依赖。

什么时候使用动态规划:

> 动态规划适合把暴力时间复杂度为指数型的问题转化为多项式的复杂度,即O(2^{n})或O(n!) 转

化为O(n^{2})

1.求最大最小的问题
2.判断可不可行,存不存在
3.统计方案个数

什么时候不用动态规划:

本来时间复杂度就在O(n^{2})或者O(n^{3})的问题继续优化,因为不怎么能优化…(100%不用DP)• 动态规划擅长与优化指数级别复杂度(2^n,n!)到多项式级别复杂度(n^2,n^3)
不擅长优化n^3到n^2

求!!所有的,具体的!!方案而不是方案个数,要用DFS而不是DP(99%不用DP), 除了N皇后
输入数据是一个集合而非序列(70~80%不用DP)，除了背包问题。

动态规划的四点要素

1.状态:即定义,中间的状态
2.方程:即从前一种状态推出现在的状态.
3.初始化:极限小的状态,即为起点.
4.答案:终点

递归三要素：
定义（状态）
接受什么参数
做了什么事
返回什么值
拆解（方程）
如何将参数变小
出口（初始化）
什么时候可以直接 return