



# Dynamic Programming

上課補充 by howard41436

# Sprout



- 影片看了嗎
- Q&A

Sprout



## 什麼是dp?

- 大家分享一下看完影片教學後，覺得什麼是dp吧！

Sprout



## 什麼是dp?

- 簡單來說，就是把大問題分成小問題
- 用小問題推出大問題的答案
- 所以建dp時我們要想兩件事
- 1. 這個問題如何切成比較簡單的小問題？
- 2. 問題的答案如何從小問題的答案推得？

Sprout



## 關於dp大家要知道的

- 很多人覺得dp很難，但其實dp是簡單化問題的方法
- 看到dp題目時先想辦法建出可以轉移的狀態就好，複雜度先擺一邊
- 定出一個不論時限的話會AC的dp演算法，往往已經是成功的一半
  - 再來剩優化，各種優化方式會在未來的dp課程教到
- dp題目多練會進步得很快，因為從題目來建立狀態的方法在很多題目中是相似的，多練就會變很強
- **想打好競賽，dp一定要強！**

Sprout



## 哪些題目可能是dp？

- dp通常拿來解決兩種問題
- 1. 最優解問題
- 2. 計數問題
- 大家想想，這兩種問題是不是很適合從小答案算出大答案呢？
- 要有這兩個性質的dp才適合dp
- 1. 重複子問題(一格只會用一次的話開陣列存起來幹嘛？)
- 2. 最佳子結構
  - 最優解問題中，要確定小問題的最佳答案跟大問題的有關
  - 計數問題中，要確定轉移來源沒有多算少算

Sprout



## dp複雜度分析

- dp複雜度分成兩個部分：
- 狀態複雜度：
  - 簡單來說 就是陣列開了多少格
  - 每格都是一個狀態，都需要算出答案
- 轉移複雜度：
  - 轉移複雜度則是 算出某一格的答案 需要的時間複雜度
  - 可以觀察轉移式來得知
- dp的總複雜度，就是總共有幾格 乘上 一格需要計算的時間！

Sprout





## dp複雜度分析

- 因為dp總是分成這兩個部分，而壓複雜度有可能是從狀態下手，也有可能是從轉移下手，因此這兩件事是要分開討論的
- 也就是“我的dp是 $n^3$ ”這句話本身不夠表示你的dp演算法，必須要說“我的dp是個狀態 $n^2$ ，轉移 $n$ ”才夠精確
- 我們通常用 $nD/mD$ 來表示一個狀態 $O(N^n)$ ，轉移 $O(N^m)$ 的dp演算法
- 在做每題dp時，都一定要好好寫出你的複雜度

Sprout





## 建構dp的小技巧

- 以下是對於常常想不到怎麼dp的同學的小技巧，不一定所有人都覺得好用
- 狀態部分：先用題目中相關的變數開好開滿，記錄所有狀態
- 轉移部分：考慮最後一格/一次可能發生的狀況，獨立開來後通常就變成前面算過的小問題了
- 如果發現轉移不了，就想想能不能把狀態開細一點，然後再回去想怎麼轉移
- 再來就是一大堆的練習題！

Sprout



## 跑步問題

- zj b589
  - 有 $n$ 段路，每段路有一個分數 $a_i$ ，你每段路可以用其中一種速度
  - 1. 用走的：你不會得到任何分數
  - 2. 用跑的：你會得到 $a_i$ 的分數
  - 3. 用衝的：你會得到 $2a_i$ 的分數，但你下一段路得用走的
- 
- 請問你最多能得到多少分？

Sprout



## 環狀最大完全不連續和問題

- acm 某round pA
- 有 $n$ 個數字圍成一環，你要選任意數量的數字，但任兩數不能相鄰，請問最大的數字總和是多少？

Sprout



## 最大連續和問題

- 影片中有最大完全不連續和問題，沒有其實更簡單的最大連續和問題，此問題常常變形出現在別的題目
- $n$ 個數字，選連續的一段數字，請問最大總和？

Sprout



## 最大不連續和問題

- npsc 2017 決賽
- $n$ 個數字，選一些數字，這些數字不能為連續的一段數字，請問最大總和？
- 狀態如何設計才能轉移好所有的可能？
- 看似簡單，但請注意細節

Sprout



## 計數dp

- UVa 11420 - Chest of Drawers
- 有一個櫃子，上到下有 $n$ 層，每層可以上鎖或不上鎖，請問讓 $s$ 個櫃子是安全的有幾種方法
- 不安全的定義：這層未上鎖或上面那層未上鎖



Sp



## 區段和

- 給你一個陣列，並且有 $q$ 次詢問，每次問某段區間 $[L, R]$ 的數字和

Sprout





## 區段和

- 給你一個陣列，並且有 $q$ 次詢問，每次問某段區間 $[L, R]$ 的數字和
- 可以利用開另外一條陣列用dp的方式紀錄第1格至第 $i$ 格的數字和，這樣要求區段和時可以用 $\text{sum}[R] - \text{sum}[L-1]$
- 這個 $\text{sum}$ 叫做前綴和陣列
- 這是非常非常常用的技巧，請大家一定要記得，看到跟區間和有關係的東西時常常可以這樣轉換
- 如果是二維的呢？（每次問你一個矩陣區塊的和）

Sprout



## 最大和矩陣問題

- 經典題
- 給你一個  $n*m$  的矩陣，所有的子矩陣中，最大的數字和是多少？
- 最裸最裸的做是  $O(n^3m^3)=O(n^6)$
- 合理的裸做是  $O(n^2m^2)=O(n^4)$

Sprout



## 最大和矩陣問題

- 經典題
- 給你一個  $n*m$  的矩陣，所有的子矩陣中，最大的數字和是多少？
- 最裸最裸的做是  $O(n^3m^3)=O(n^6)$
- 合理的裸做是  $O(n^2m^2)=O(n^4)$

提示：能不能想辦法讓這題跟最大連續和扯上關係？

Sprout



## 矩陣最大空方型問題

- 給你一個01矩陣，請問裡面最大的全部都是0的方形有多大？

Sprout



## 矩陣最大空矩型問題

- 給你一個01矩陣，請問裡面最大的全部都是0的矩形有多大？

Sprout



## 矩陣乘法問題

- 給你一系列矩陣，要從第一個乘到最後一個，保證兩相臨矩陣之間都是可以做乘法的
- 一個  $a*b$  的矩陣乘上一個  $b*c$  的矩陣需要做  $a*b*c$  次數字的乘法
- 矩陣有結合律，所以在不調換矩陣順序的狀況下可以用任意順序做乘法
- 請問把所有矩陣乘起來最少需要做幾次數字乘法？

Sprout



## 矩陣乘法問題

- 這種dp是前面沒遇到過的，前面遇到的題目都是按照前到後的順序做，於是我們的狀態設計可以很容易的從前到後轉移
- 這種可以按照任意順序去操作的題目，怎麼設計狀態？

Sprout





## 矩陣乘法問題

- 由於序列本身的順序不能改變，因此如果我們以某個區間當成狀態，通常可以從左右的子區間轉移答案
- 例如本題， $[L, R]$ 的矩陣要全部乘在一起，一定是先把 $[L, K]$ 的矩陣乘在一起，以及 $[K+1, R]$ 的矩陣乘在一起，再把剩下兩個矩陣相乘
- 枚舉 $K$ 後，剩下的部分是子問題
- 狀態：2D
- 轉移：1D

Sprout



## 矩陣乘法問題

- zj d652 : 題目中沒出現矩陣, 但其實就是矩陣乘法問題
- UVa 00348: 矩陣乘法, 要印出最佳解的時候乘的順序
- Case 1:  $(A1 \times (A2 \times A3))$
- dp時記錄最佳的轉移來源, 最後就能循序印出答案了!
- 類似經典題: Optimal Binary Search Tree
- 有興趣的人可以查查看這是什麼題目, 然後想想看怎麼做
- 那我們接下來再來作一題區間的dp

Sprout



## 消消樂

- UVa 10559
  - 有一排方塊，每個方塊都有顏色
  - 每次可以把連續顏色的一段消掉，得到(消去長度)<sup>2</sup>的分數
  - 請問全部消完最多可以得到幾分？
- 
- 跟剛剛那題很像的感覺，大家列列看狀態和轉移式吧！

Sprout



## 消消樂

- UVa 10559
  - 有一排方塊，每個方塊都有顏色
  - 每次可以把連續顏色的一段消掉，得到(消去長度)<sup>2</sup>的分數
  - 請問全部消完最多可以得到幾分？
- 
- 跟剛剛那題很像的感覺，大家列列看狀態和轉移式吧！
  - 你確定你的演算法是對的嗎？

Sprout