



Greedy

Lecture by **erichung**

Credit: qazwsxedcrfvtg14, boook

Sprout



影片複習

Sprout



What is Greedy?

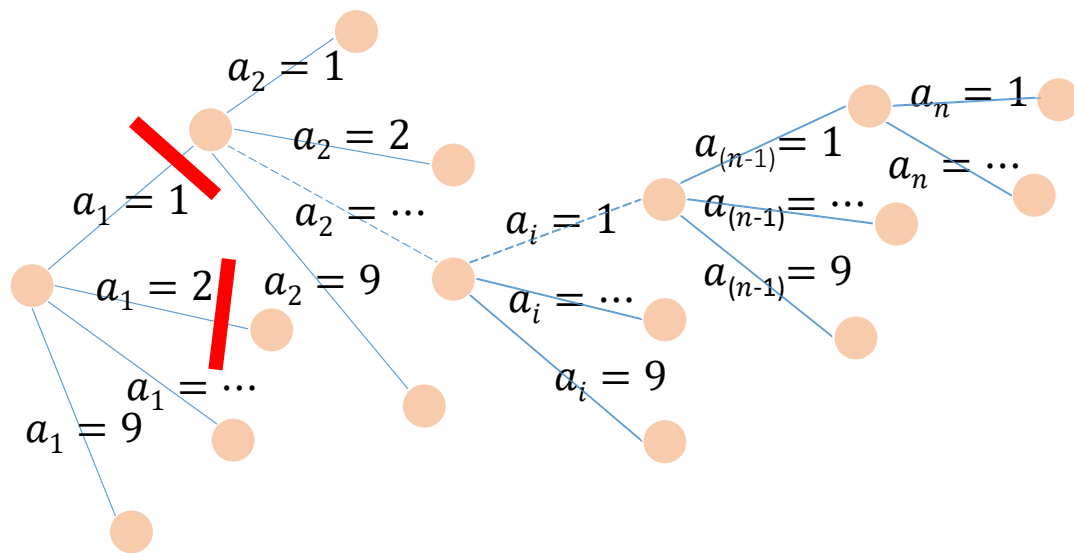
- 做出對於當前情況下最佳的策略
(並希望他最終能導向最佳解)

Sprout



Greedy應該要怎麼想 - 枚舉在幹麼？

- 枚舉事實上就是試圖拜訪整棵決策樹，找出樹裡的解節點
 - 剪枝事實上就是根據一些推理排除絕對不可能的分支



5	8	a_1
2	a_2	7
3	a_3	1

- 有時候，我們可以根據一些問題的特性，確定解節點所在的分支！
 - 總是選擇當前看起來最有利的分支，然後義無反顧

Sprout



各式各樣的題目

Sprout



簡單的題目(嗎?): Add All

黑板上有 N 個數字, 你每次可以選兩個數字 a, b , 把它們擦掉後, 再把 $a+b$ 寫到黑板上, 這麼做需要花 $a+b$ 的成本。你做這件事 $N-1$ 次後會剩下一個數字, 請問至少要花多少成本。

Example: $[1, 2, 3]$

$[1, 2, 3]$	\Rightarrow	$[3, 3]$	\Rightarrow	$[6]$	cost: 9
$[1, 2, 3]$	\Rightarrow	$[2, 4]$	\Rightarrow	$[6]$	cost: 10
$[1, 2, 3]$	\Rightarrow	$[1, 5]$	\Rightarrow	$[6]$	cost: 11

Sprout



這不是很簡單嗎

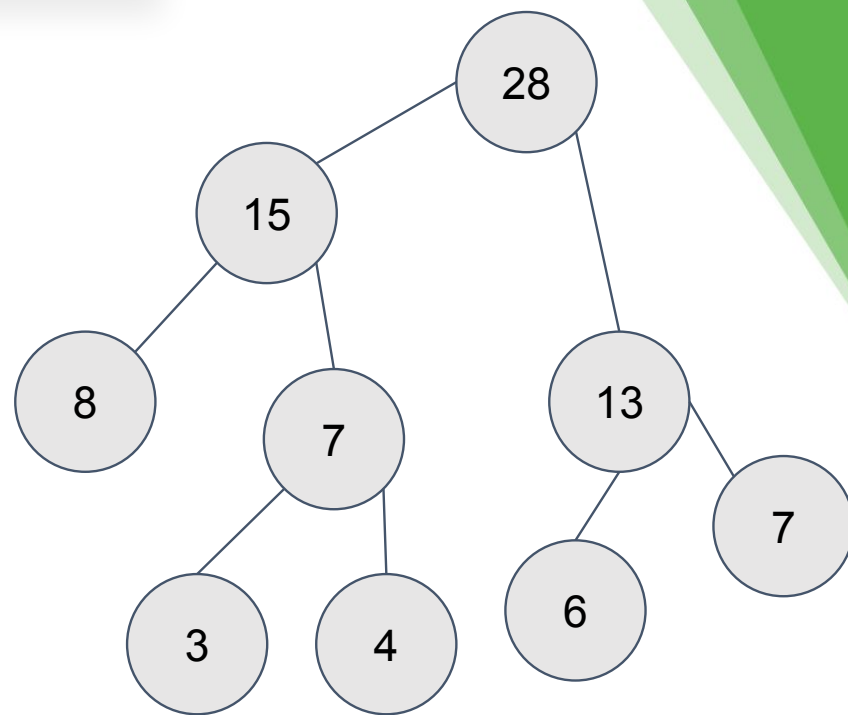
- 既然我把兩個數字合起來的成本和它們的大小有關，那是不是我總是選兩個最小的合起來就好
- 聽起來很有道理，可是為什麼？

Sprout



好像也沒那麼簡單

- 如果合併的過程是一棵樹...
- 好像有點眼熟？
- 這不是霍夫曼編碼嗎



Sprout



不那麼顯然：物品堆疊

有 N 個物品，其中第 i 個的重量是 $w(i)$ ，要被使用的次數是 $f(i)$ 。你要決定某個順序 a_1, a_2, \dots, a_N 由上而下把它們疊起來。接下來由上而下第 k 個物品會被取用 $f(a_k)$ 次，每次被取用的時候，都要把它上面的全部物品（不包含它自己）拿起來，所以總共會花上

$$f(a_k) * \sum_{i=1..k-1} w(a_i)$$

那麼多的力氣。求總力氣最少可以是多少。



直覺的想法

- 越上面的東西，重量會被算越多次，那就把重的放下面好了
 - $f=[1,100]$ ， $w=[1,2]$
- 那我把用比較多次的放上面總行了吧
 - $f=[2,1]$ ， $w=[100,1]$
- 怎麼不管怎麼排都是錯的 QQ

Sprout



有邏輯的思路

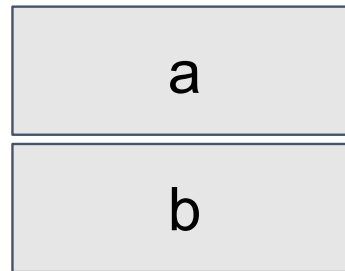
- 單靠亂猜猜出答案似乎不太可能
- 還記得「誰先晚餐」的證明嗎？
- 交換兩個相鄰的物品會發生什麼事？

Sprout

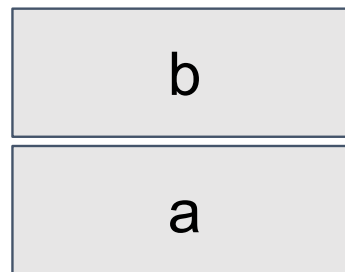


有跡可尋

- 把兩個相鄰物品 a, b 交換時，並不會改變其他物品被拿起的次數，以及它們兩個因為其他物品被取用而被拿起的次數，所以我們就只要考慮它們之間的影響就好
- 要是 $w(a) * f(b) > w(b) * f(a)$ 那把它們交換後一定可以拿到一個更好的答案



$$w(a) * f(b)$$



$$w(b) * f(a)$$

Sprout



然後...

$$w(a) * f(b) \leq w(b) * f(a)$$

- 要怎麼做出一個對於相鄰的兩個物品, 上面是 a 下面是 b, 都滿足這個條件的順序啊?
- 試著移項一下...

$$w(a)/f(a) \leq w(b)/f(b)$$

Sprout



CSES - Tasks and Deadlines

- 你要執行 n 個任務，每個任務都有需要花費的時間以及死線，
- 你可以任意決定你要執行的順序，執行完一個任務後，你可以獲得獎勵，獎勵是 $f - d$ ， f 是執行完該任務的時間 d 是他的死線。
- 請最大化獎勵。

Sprout



CSES - Tasks and Deadlines

- 對於一個任務 $d_i - f_i$
- d_i 是固定的所以可以不用考慮他
- 目標：找到一個順序使得所有任務的結束時間的總和最小
- Greedy-choice: 選擇花費時間最少的任務先做

Sprout



Greedy choice property

- 反證法
- 考慮有一個最佳解使得花費時間最少的任務不是第一個執行的
- 我們可以將花費時間最少的任務跟第一個執行的任務交換順序
- 那這樣對於所有在原本第一個任務和花費時間最少的任務之間的所有任務都會有比較小的結束時間，其他任務則不變
- 因此這樣交換會有更好的解
- 與假設矛盾 得證

Sprout



Greedy choice property

- 4 個任務且他們的花費時間是 $\{3, 4, 5, 6\}$
- 執行順序：5 \rightarrow 4 \rightarrow 3 \rightarrow 6
結束時間：5, 9, 12, 18
- 將 3 和 5 交換：3 \rightarrow 4 \rightarrow 5 \rightarrow 6
結束時間：3, 7, 12, 18 (**better!**)

Sprout



先入為主: Programmers and Artists

有 n 個數對 (x, y) , 你要決定每個數對是 type 1 還是 type 2, type 1 的數對會提供 x 的價值, type 2 的會提供 y 的價值, 你要有恰好 a 個 type 1 和 b 個 type 2 ($a+b=n$), 求最大總價值是多少。



CSES 2426

Sprout



先入為主

- 題目根本不用那麼複雜吧，我就當成每個數對一開始都是 type 1，再挑 b 個變成 type 2 就好了
- 一個數對從 type 1 變成 type 2，總價值會多出 $y-x$
- 那就選 $y-x$ 最大的 b 個數對變成 type 2
- 改變問題來讓需要做的決定變簡單

Sprout



亞可的問題

- 路上總共有 $N + M$ 張卡片，每張卡片可能都有不同的價值。
- 現在想要選擇 N 張卡片帶回家
- 因為特殊的因素，你希望在留在地上的卡片的價值平均加上帶回家的卡片的價值平均最小。
- 請問這個平值最小可以多小？

Sprout



亞可的問題

- 路上總共有 $N + M$ 張卡片，每張卡片可能都有不同的價值。
- 現在想要選擇 N 張卡片帶回家
- 因為特殊的因素，你希望在留在地上的卡片的價值平均加上帶回家的卡片的價值平均最小。
- 請問這個平值最小可以多小？
- 題目可以等價轉換為：
 - 選擇 N 張卡片把價值 $\ast (1 / N)$
 - 選擇 M 張卡片把價值 $\ast (1 / M)$

Sprout



奇怪的決策：雲霄飛車

有 n 根柱子，高度是 h_1, h_2, \dots, h_n ，你每次可以交換兩根相鄰的柱子，求你至少要交換幾次才能使得存在一個 k ，滿足

$$h_1 \leq h_2 \leq \dots \leq h_{k-1} \leq h_k \geq h_{k+1} \geq \dots \geq h_{n-1} \geq h_n$$

也就是這些柱子是中間高兩邊低。

2019 TOI 初選 pD

Sprout



不同的決策順序

- 先決定第一根柱子是誰？
- 沒什麼想法...
- 不如我們先不要想「第一根柱子是誰」
而是先想「某根柱子要在哪」
- 最矮的柱子一定在最邊邊！

Sprout



不同的決策順序

- 看看最矮的柱子離哪邊比較近，直接把它交換到那邊去
- 再此之後我們再也不用理它了
- 等等真的嗎，要是最矮的柱子有很多根怎麼辦
- 你不會想要兩根最矮的柱子交換位置，
所以你可以想像這些最矮的柱子一起消失了
 - 一根柱子所花的交換次數，其實是「它左邊或右邊比它高的柱子數」
- 這樣可以得到一個 $O(N^2)$ 的演算法，
搭配一些資料結構可以做到 $O(N \log N)$

Sprout



決策順序的差別

- 決策樹其實不單單只是「把所有可能的決策列出來」那麼簡單而已
- 以這題為例，要是你做決定的順序是決定第一根柱子、決定第二根柱子、... 那你很難找出 Greedy 的策略
- 改變一下做決定的順序，世界可能會不一樣
 - 「某件事決定好後就可以假裝它不存在」是個不錯的切入點

Sprout



CSES - Towers

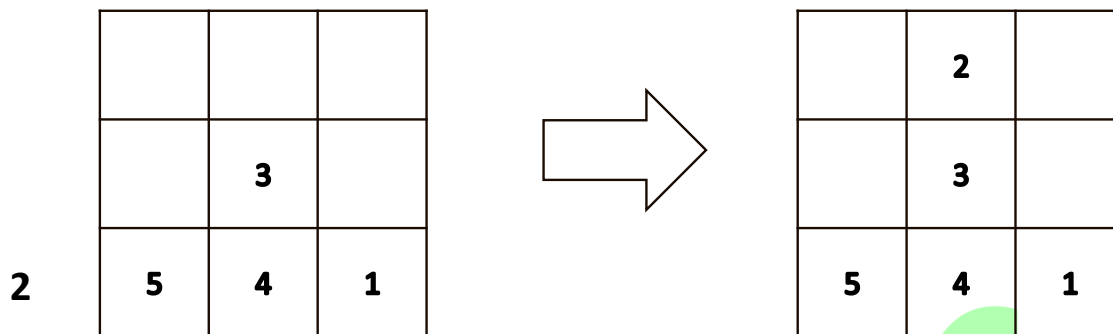
- 你有 n 個方塊，每個方塊都有大小，要把他們疊成塔且每一座塔的大小由低到高必須是大到小
- 你需要依照給定順序去處理這 n 個方塊
- 對於每個方塊，你可以放到某個方塊的上面，或者是自己另外新增一座塔
- 請問塔的數量最少是多少

Sprout



CSES - Towers

- 每次將方塊放到所有可放的塔中，最小的上面，如果不存在可以放的位置就自己另成一座新塔



Sprout



CSES - Towers

- 假設存在一組最佳解沒有遵守我們的 Greedy 策略
- $y < x_1 < x_2$
- 最佳解把 y 放在 x_2 上
- 我們需要證明把 y 放在 x_1 上也能是一組最佳解

	x_2		
		x_1	
INF	INF	INF	INF

Sprout



CSES - Towers

- 我們要證明，將 y 改放在 x_1 上仍然存在一種方法可以堆出最佳解
- 證明，可以將所有在 x_2 以上的元素跟 x_1 以上的元素交換
- 因為 $y < x_1$ 且 $x_1 < x_2$ ，這樣交換之後仍然是合法的
- 仍然是一組最佳解
- 對於沒有遵守 Greedy-choice 的最佳解，我們可以把它轉換成有遵守 Greedy choice 的解且仍然是最佳的

	
	y	...	
	x_2	...	
		x_1	
INF	INF	INF	INF

Sprout



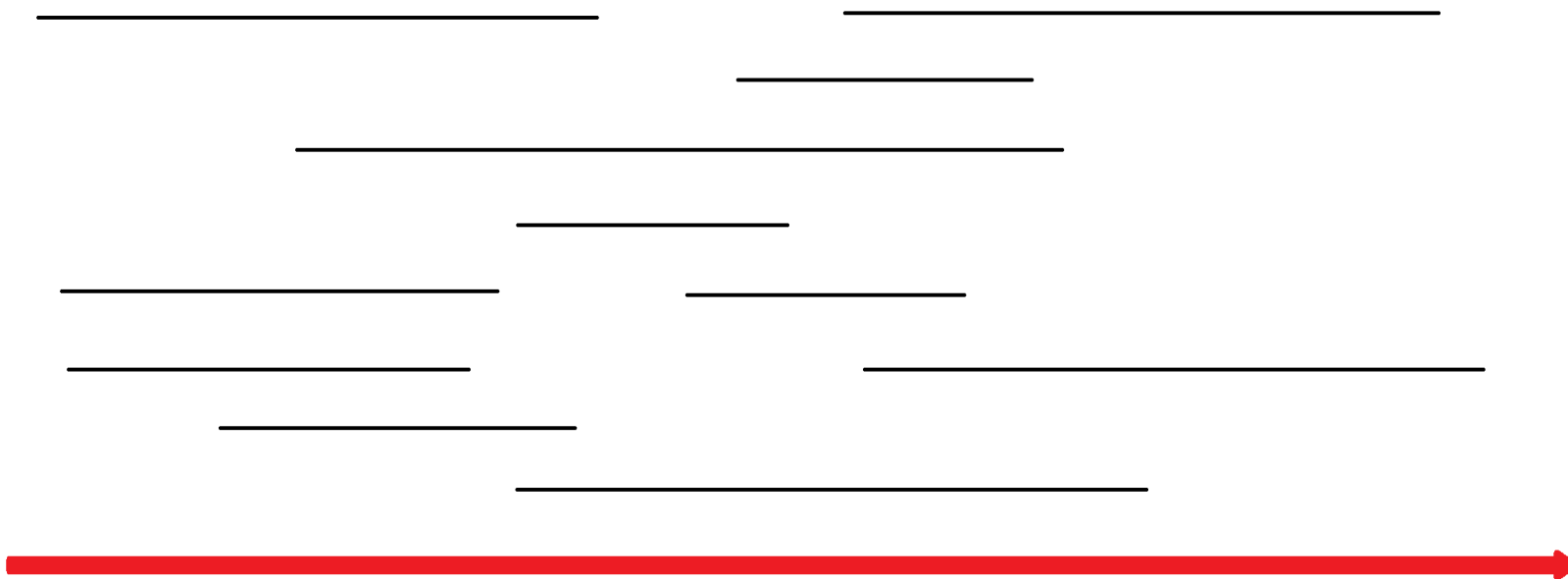
線段三部曲V2 -51nod 1091

- 題意：不重疊的線段
 - X 軸上有 N 條線段，每條線段包括起點和終點。
 - 現在起點或終點重疊，**不算重疊**
 - 線段的重疊計算方式：
 - $[2, 3]$ 和 $[3, 6]$ **不算重疊**。
- 請從 N 條線段中選出最多條彼此不相交的線段
- $2 \leq N \leq 10000$

Sprout



選擇最多不重疊的線段



Sprout



可能的 Greedy 方法

- 想出一些 Greedy 的方案：
 - 優先選擇最短的線段
 - 優先選擇和線段最少重疊的線段
 - 優先選擇比較早開始的線段
 - 優先選擇比較早結束的線段

Sprout



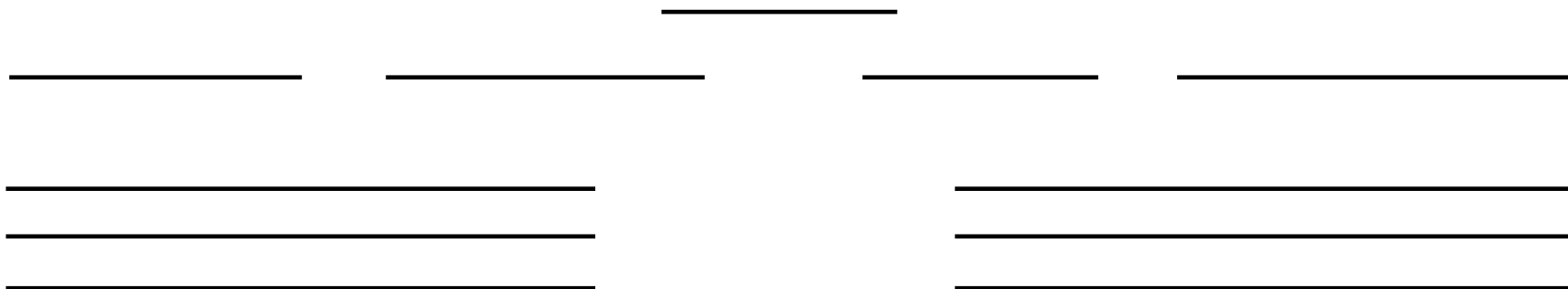
1. 優先選擇最短的線段-WA



Sprout

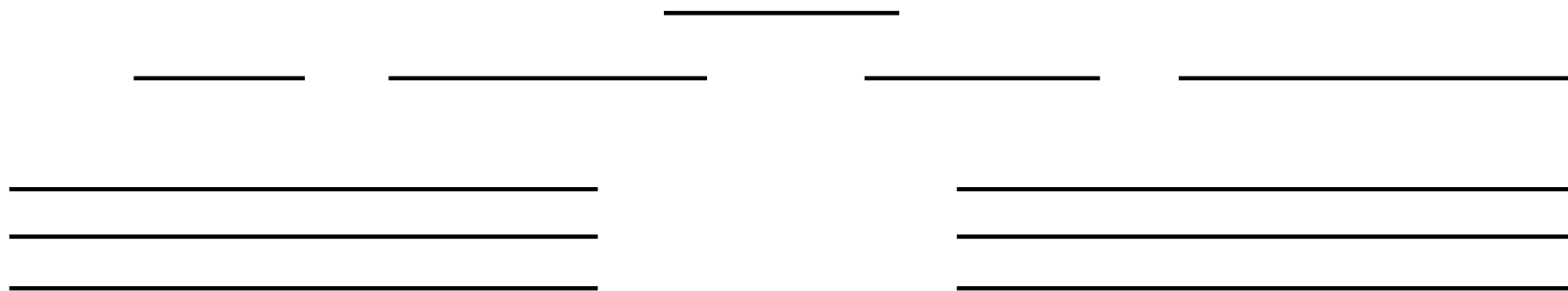


2. 優先選擇和線段最少重疊的線段-WA





3. 優先選擇比較早開始的線段-WA

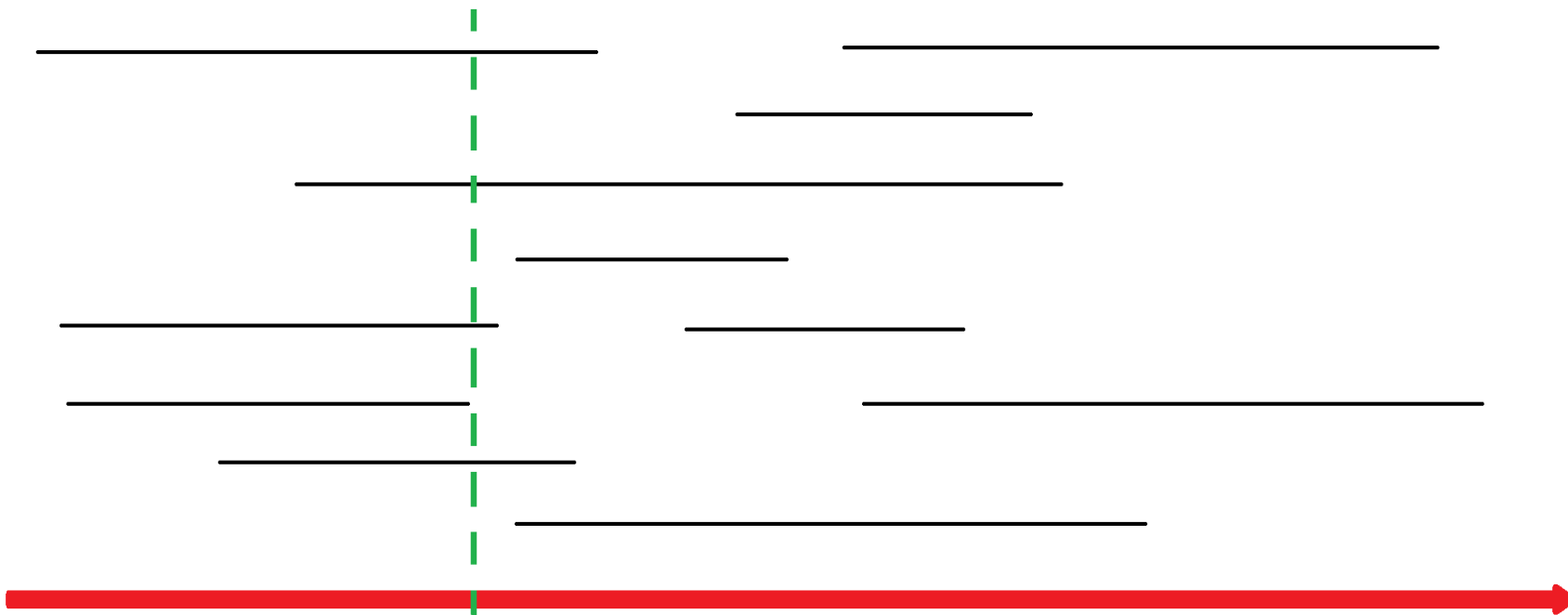


aprot



4. 優先選擇比較早結束的線段-AC

- 感性的想法：在「最短」的時間內得到一條線段





一層包裝：基地台

有一條數線，上面有 N 個給定的點，你最多可以設置 K 座基地台，並且決定一個數字 D ，使得對於每一個給定的點，離它最近的基地台距離都不超過 $R=D/2$ 。求 D 至少要是多少。

APCS 20170304 p4

Sprout



拆一層包裝

- 若我們直接想辦法解出對於 D 的最佳化問題，題目好像有點太複雜了
- 還記得影片裡的抄書問題嗎？
- 對 D 二分搜！
- 接下來的問題是：
在 D 固定的情況下，我可以只設 K 座基地台嗎？

Sprout



最佳化問題與判定問題

- 我們就這麼麼把一個最佳化問題變成判定問題了！
- 抄書問題拆完包裝後，很明顯根據拆出來的問題的定義，只要讓每個人都抄不超過上限的數量並盡量多就好
- 怎麼設基地台？

Sprout



總算可以 Greedy

- 從最左邊的給定點開始枚舉
- 假設這個點的位置是 x ,
要是目前它不被任何基地台的服務範圍蓋到,
那就在 $x+R$ 處設一個基地台(服務範圍涵蓋到 $x+D$)
- 其實我們在做的事情是找「 D 固定時至少要設幾座基地台」
- 不得不這麼做才去做
 - 拖延也是一種貪心 (?)

Sprout



Greedy應該要怎麼想 – 小技巧？

- 三個思考的小方向：
 - 沒有理由不這樣做：直接選一個選項，並確保不這樣選不會比較好
 - 可能這是最佳的選擇
 - 可能這個選擇不會影響到選擇最好的答案
 - 檢查：如果答案長這樣，那這個答案會被考慮到
 - 想想所有可能的情況，看看答案是不是都會被想到
 - 從反方向想想看
- 比賽中，乍看不是錯的可能就會是對的
 - 容易實作的話也可以直接寫寫看

Sprout