

貪心法&分治法



分治法

(Divide and Conquer)

分而治之

實作上在一層遞迴中有三個步驟：

1. 分解：將問題切成更小的子問題
2. 解決：當問題夠小時直接求解
3. 合併：把所有子問題的答案合併成問題的答案

快速幂

求 x^y ?

1. 分解：當 y 是偶數，分成 $x^{\frac{y}{2}}$ 和 $x^{\frac{y}{2}}$ 當 y 是奇數，分成 $x^{\lfloor \frac{y}{2} \rfloor}$ $x^{\lfloor \frac{y}{2} \rfloor}$ x
2. 解決：當 y 是 0 答案為 1
3. 合併：將子問題的答案乘起來

時間複雜度： $O(\log(y))$

快速幂 - 程式碼

```
int fast_power(int x, int y) {  
    if(y == 0) return 1;  
    int k = fast_power(x, y/2);  
    if(y % 2) return k * k * x;  
    else return k * k;  
}
```

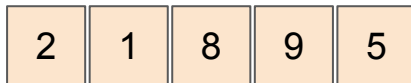
合併排序法

排序長度 n 的數列 x ?

1. 分解：把數列剖半成**左子數列**和**右子數列**
2. 解決：當數列只剩一個元素，什麼都不用做
3. 合併：左右子數列已經從左到右排序好，從兩個子數列中一個個選出最小的元素出來排列即為答案

時間複雜度： $O(n\log(n))$

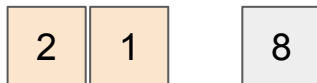
合併排序法 - 詳細圖解



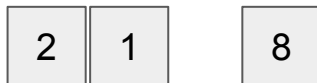
合併排序法 - 詳細圖解



合併排序法 - 詳細圖解



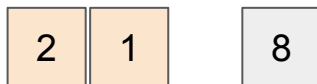
合併排序法 - 詳細圖解



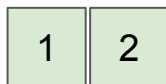
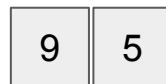
合併排序法 - 詳細圖解



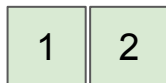
合併排序法 - 詳細圖解



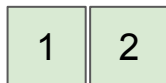
合併排序法 - 詳細圖解



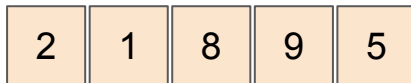
合併排序法 - 詳細圖解



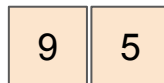
合併排序法 - 詳細圖解



合併排序法 - 詳細圖解



合併排序法 - 詳細圖解



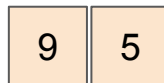
合併排序法 - 詳細圖解



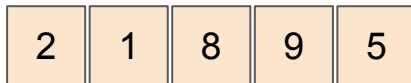
合併排序法 - 詳細圖解



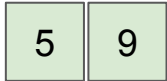
合併排序法 - 詳細圖解



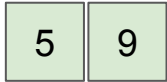
合併排序法 - 詳細圖解



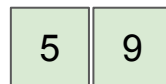
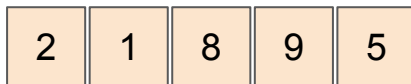
合併排序法 - 詳細圖解



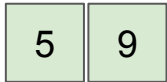
合併排序法 - 詳細圖解



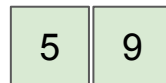
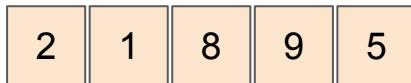
合併排序法 - 詳細圖解



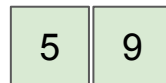
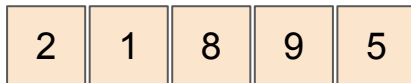
合併排序法 - 詳細圖解



合併排序法 - 詳細圖解



合併排序法 - 詳細圖解



合併排序法 - 詳細圖解

1	2	5	8	9
---	---	---	---	---

DONE

最近點對

找 n 個點的最近點對？

1. 分解：將 n 個點根據 x 軸分成左半邊 $n/2$ 個點和有半邊 $n/2$ 個點
2. 解決：只剩兩個點時，答案就是兩個點的距離
3. 合併：答案只有三種可能，**左半邊的點對**，**右半邊的點對**，**跨兩半邊的點對**，找出跨兩半邊的點對，把三種可能的答案取 \min

時間複雜度： $O(n\log(n))$

最近點對 - 合併

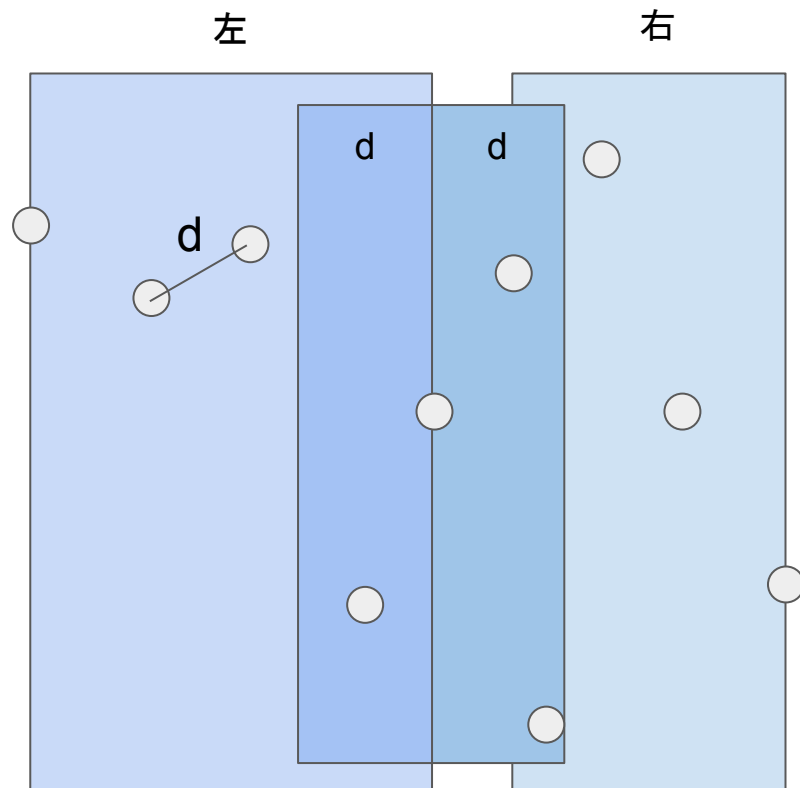
假設左右兩半邊的答案取最小是 d

代表左半邊的所有點對和右半邊的所有點對的距離都 $\geq d$

以左半邊的右邊界為標準左右手個張開 d , 範圍內是可能產生新的最近點對的點

接著暴搜所有目標區左邊點和右邊點的組合

最近點對 - 合併



注：點沒有大小

最近點對 - 思考

有比暴搜所有目標區左邊點和右邊點的組合還好的方式嗎？



最近點對 - 題目

UVA 10245

UVA 10750

UVA 11378

Codeforces 429D



貪心法 (Greedy)

貪心性質：

局部的最佳解構成全部的最佳解

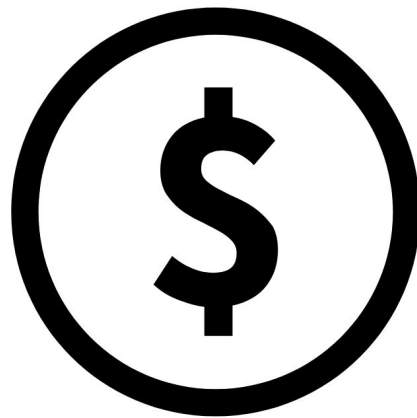
最佳子結構：

問題的最佳解包含子問題的最佳解

最小硬幣問題 - 問題敘述

硬幣幣值有 1, 5, 10 三種

問要湊出 x 元, 最少需要幾個硬幣?



最小硬幣問題 - 貪心法

要湊出 x 元我們一次選一個硬幣

有三種選擇: 1, 5, 10

選擇 $\leq x$ 中**最大**的幣值 c

選完我們剩下一個更小的子問題: 湊出 $x-c$ 元

最小硬幣問題 - 思考

滿足貪心性質？是否選擇最大幣值的硬幣最好？

在幣值 1, 5, 10 的情況下，具有貪心性質

因為如果我們要最小化硬幣的個數

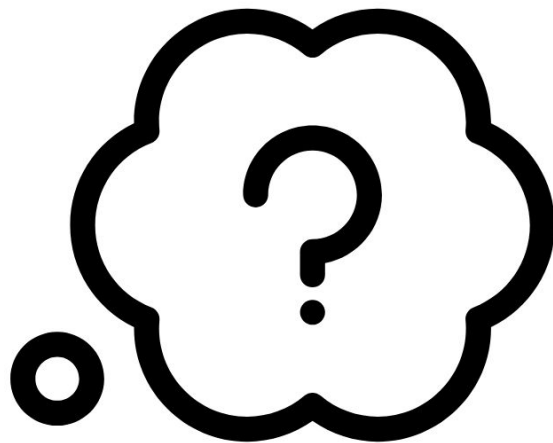
1 元的個數就會 < 5 個 (當 1 元的個數 ≥ 5 時，換成 5 元會更好)

5 元的個數就會 < 2 個 (當 5 元的個數 ≥ 2 時，換成 10 元會更好)

只用 1 元和 5 元只能湊出 0 - 9 元，要湊出 ≥ 10 元的話一定要換 10 元直到小於 10 元 (就是選擇最大幣值直到不能再選)

最小硬幣問題 - 思考

如果我們的硬幣有 1, 5, 6, 9 還可以用貪心法來求解嗎？



最小硬幣問題 - 參考文件

有一個 $O(N^3)$ 的算法可以驗證這組硬幣是否可以用貪心法求解

演算法筆記：

<http://www.csie.ntnu.edu.tw/~u91029/KnapsackProblem.html#7>

論文：

<https://ecommons.cornell.edu/bitstream/handle/1813/6219/94-1433.pdf?sequence=1&isAllowed=y>

霍夫曼編碼 (Huffman Coding)

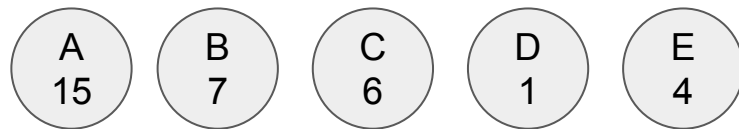
一種編碼方式

出現次數越高的字母的編碼長度越短，讓文本的總長度最短

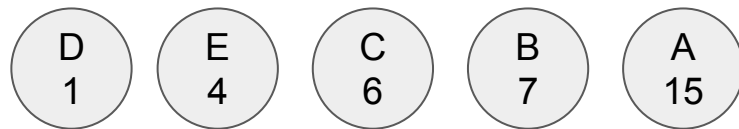
使用貪心法：

每次選最小的兩個頻率的節點合併成新的節點，直到只剩一個

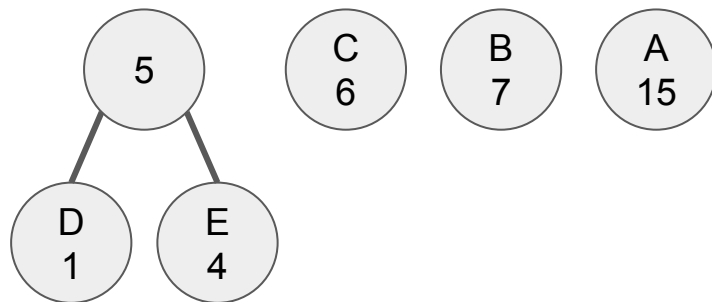
霍夫曼編碼 (Huffman Coding)



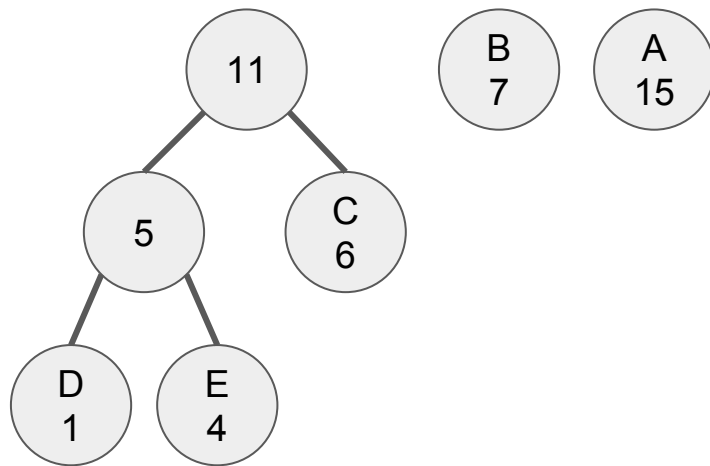
霍夫曼編碼 (Huffman Coding)



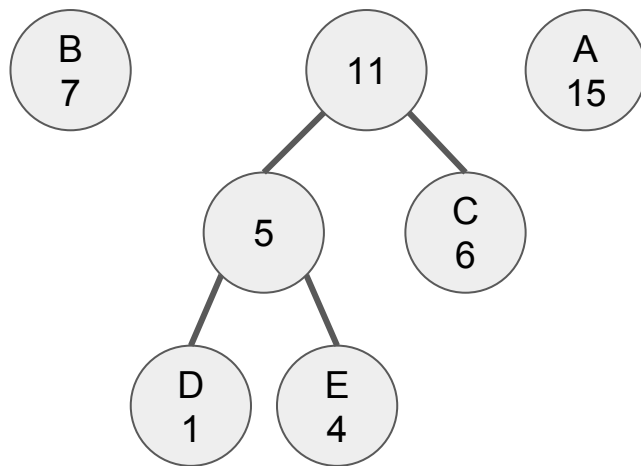
霍夫曼編碼 (Huffman Coding)



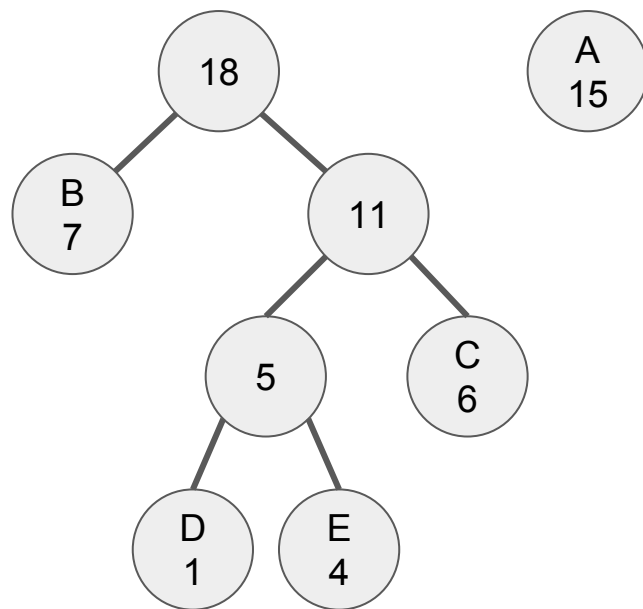
霍夫曼編碼 (Huffman Coding)



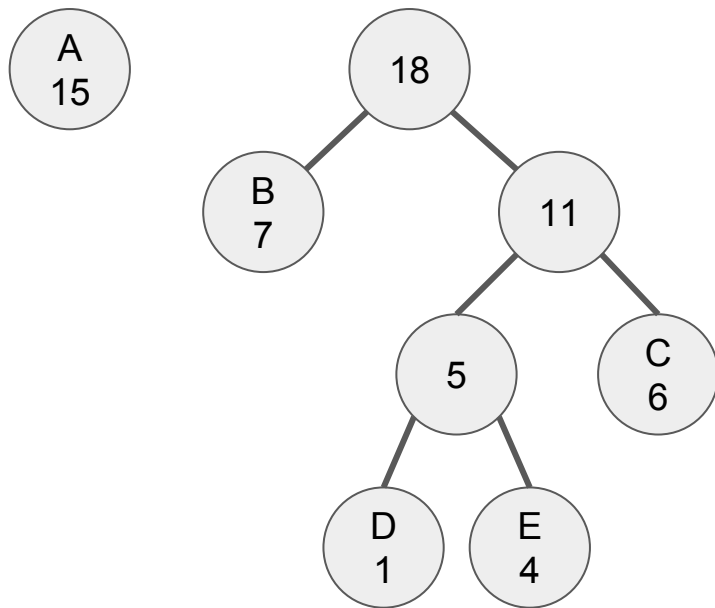
霍夫曼編碼 (Huffman Coding)



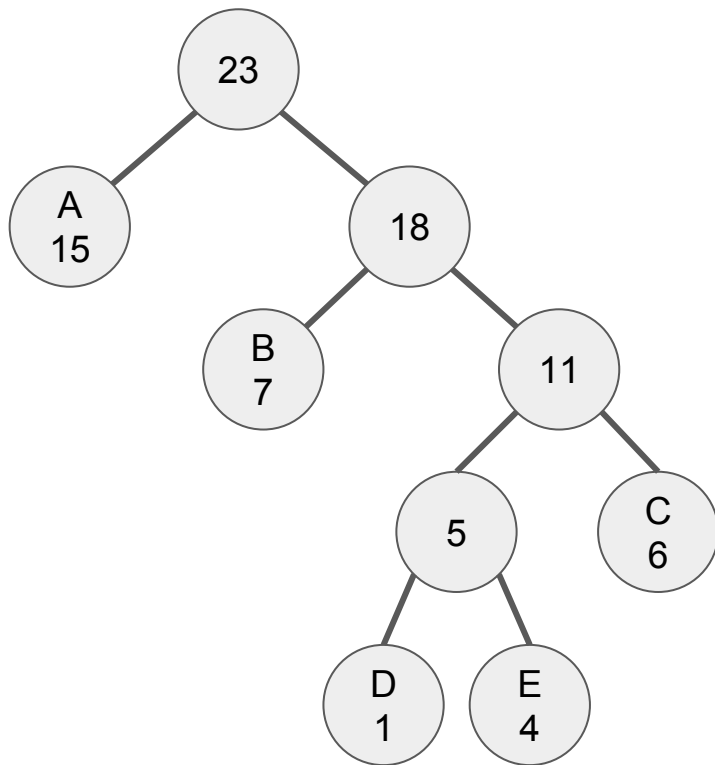
霍夫曼編碼 (Huffman Coding)



霍夫曼編碼 (Huffman Coding)



霍夫曼編碼 (Huffman Coding)



霍夫曼編碼 (Huffman Coding)

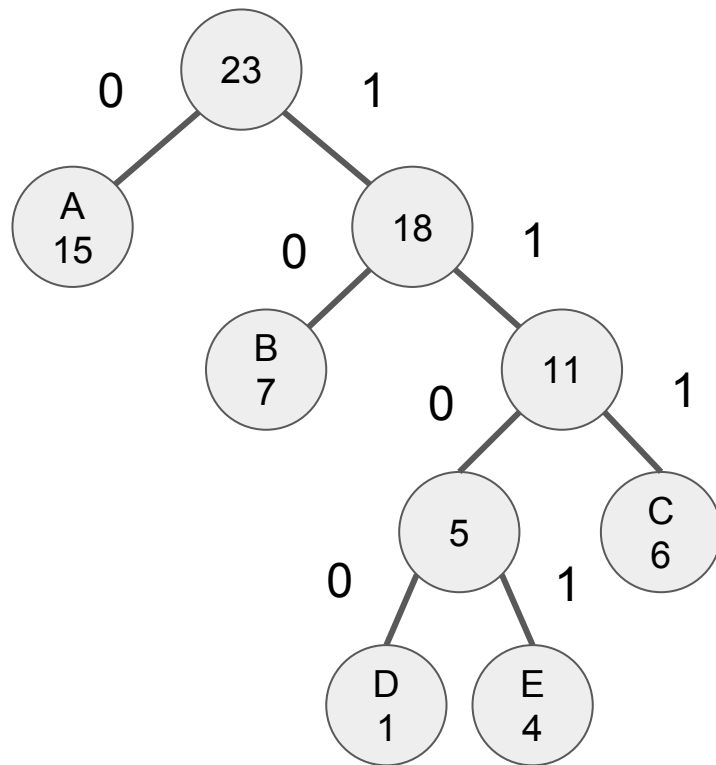
A : 0

B : 10

C : 111

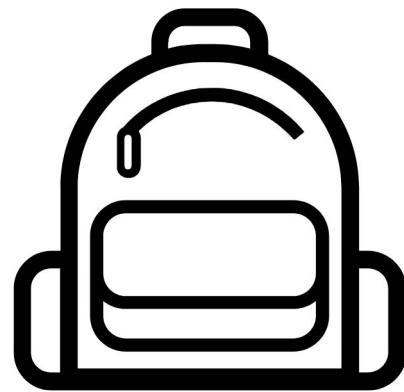
D : 1100

E : 1101



小數背包 - 題目敘述

有一個背包容量為 C ，要裝 n 個物品，物品 i 價值 V_i 佔據容量 W_i ，可以只放置部分物品（物品可切割），求背包能裝的最大價值？



小數背包 - 貪心法

假設我們一定能將背包裝滿（裝不滿答案就是全部的物品）

那每單位容量的價值越高我們的總價值越高

每次都選**每單位容量價值最高的**放，直到放滿

價值	25	20	15
容量	18	15	10
價值/容量	1.389	1.333	1.5

小數背包 - 思考

如果物品只能完整地放呢？是不是就不能用貪心法了？

