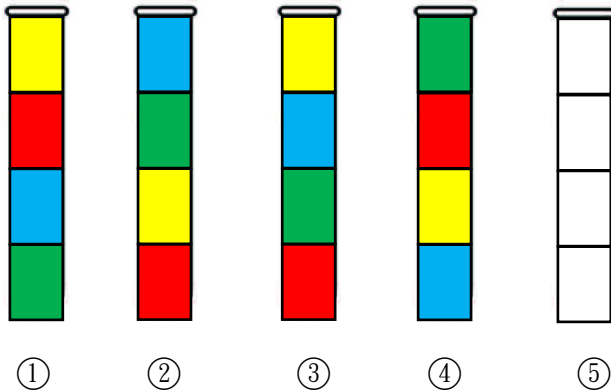


水中自有顏如虹 四 年級 姓名：黃詠恩 資優班座號：23

有一天高愛迪斯看見了一杯色彩繽紛的漸層飲料覺得實在太神奇了，當下就決定買三杯來試試，果然清涼可口，這時突然靈光一閃，「如果限定只能把相同顏色的飲料倒在一起，在只有1瓶或2瓶空瓶的情況下，可不可以最終把所有飲料都倒成相同顏色？」這果然是一個有趣的數學問題，各位聰明的小朋友你們可以幫高愛迪斯解決這個難纏的數學問題嗎？



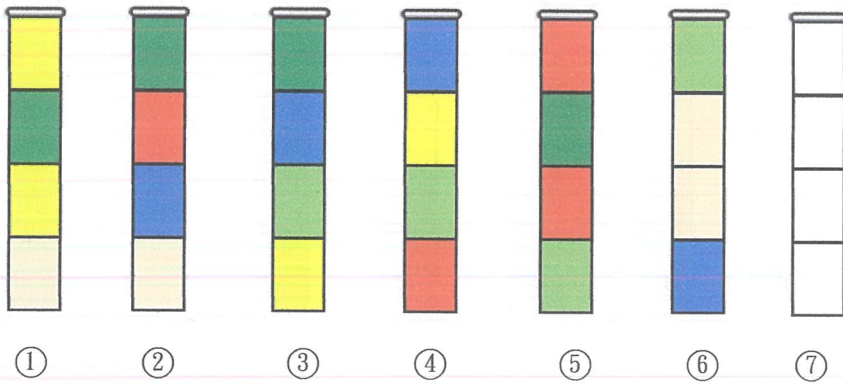
【例題】下面有4個填滿4種不同顏色的試管，以及一個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。



解題紀錄：

第1次	①⇒⑤	第6次	④⇒⑤	第11次	①⇒④
第2次	③⇒⑤	第7次	③⇒④	第12次	③⇒①
第3次	②⇒③	第8次	②⇒③	第13次	③⇒②
第4次	④⇒②	第9次	②⇒⑤	第14次	
第5次	④⇒①	第10次	①⇒②	第15次	

【第一關】下面有6個填滿4種不同顏色的試管，以及1個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。



解題顏色使用說明：

	R(紅 Red)		Y(黃 Yellow)		Bk(黑 Black)		Pk(粉 Pink)
	G(綠 Green)		LtY(淺黃 LtYellow)		S(銀 Silver)		
	LtB(淺藍 LtBlue)		P(紫 Purple)		Gy(灰 Gray)		
	LtG(淺綠 LtGreen)		B(藍 Blue)		O(橘 Orange)		

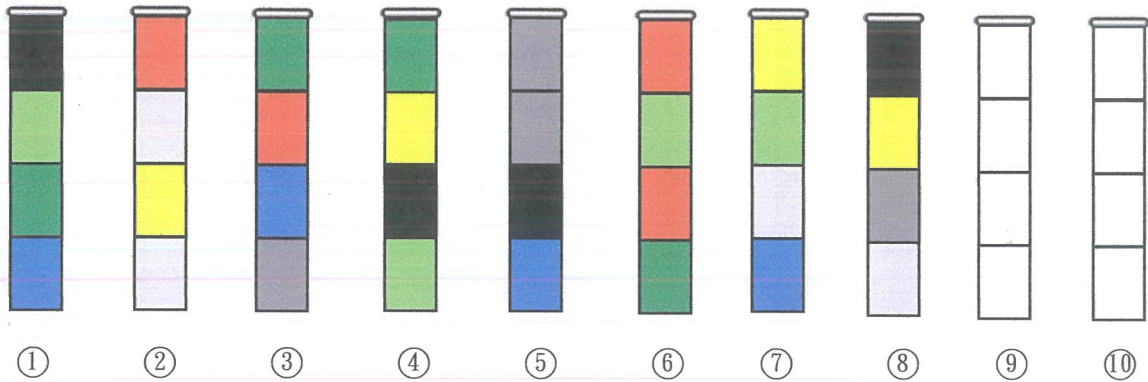
解題紀錄：說明：(Bx2)表示藍色一次倒2層，(Bx3)表示藍色一次倒3層，(B)表示藍色1層

第1次	(G) → 7	第6次	(R <sub>x2</sub> ) → 5	第11次	(Y <sub>x2</sub> ) → 1	第16次	(LtY) → 1
第2次	(G) → 7	第7次	(LtB <sub>x2</sub> ) → 2	第12次	(LtG <sub>x2</sub> ) → 4	第17次	(LtG <sub>x3</sub> ) → 2
第3次	(LtB) → 3	第8次	(LtG) → 3	第13次	(Y <sub>x3</sub> ) → 3	第18次	(R <sub>x3</sub> ) → 4
第4次	(R) → 2	第9次	(Y) → 4	第14次	(LtY <sub>x2</sub> ) → 1	第19次	(LtG <sub>x5</sub> ) → 5
第5次	(G) → 7	第10次	(G) → 7	第15次	(LtB <sub>x3</sub> ) → 6	第20次	→

最終結果：

試管 1	LtYellow	LtYellow	LtYellow	LtYellow	
試管 2	空				
試管 3	Yellow	Yellow	Yellow	Yellow	
試管 4	Red	Red	Red	Red	
試管 5	LtGreen	LtGreen	LtGreen	LtGreen	
試管 6	LtBlue	LtBlue	LtBlue	LtBlue	
試管 7	Green	Green	Green	Green	

【第二關】下面有8個填滿4種不同顏色的試管，以及2個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。(1★)



解題紀錄：

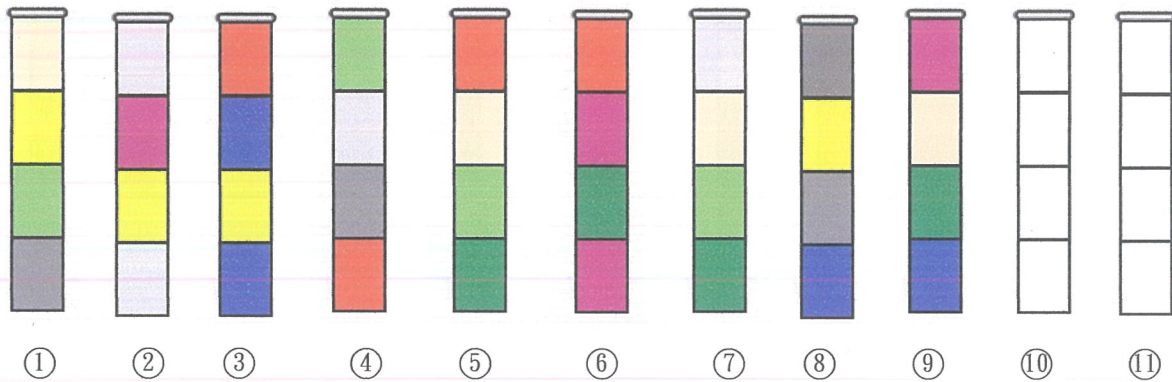
第 1-7 次	3 → 9 (G)	3 → 10 (R)	4 → 9 (G)	6 → 10 (R)	7 → 4 (Y)	6 → 7 (LtG)	6 → 10 (R)
第 8-14 次	6 → 9 (G)	4 → 6 (Yx2)	8 → 4 (Bk)	8 → 6 (Y)	5 → 8 (Gyx2)	4 → 5 (Bkx2)	7 → 4 (LtGx2)
第 15-21 次	2 → 10 (R)	7 → 2 (S)	3 → 7 (LtB)	8 → 3 (Gyx3)	2 → 8 (Sx2)	2 → 6 (Y)	8 → 2 (Sx3)
第 22-28 次	5 → 8 (Bkx3)	5 → 7 (LtB)	1 → 8 (Bk)	1 → 4 (LtG)	1 → 9 (G)	7 → 1 (LtBx3)	→
第 29-35 次	1 → (LtBx3)	→	→	→	→	→	→

最終結果：

試管 1	LtBlue	LtBlue	LtBlue	LtBlue	Blue
試管 2	Silver	Silver	Silver	Silver	Silver
試管 3	Gray	Gray	Gray	Gray	Gray
試管 4	LtGreen	LtGreen	LtGreen	LtGreen	Light Green
試管 5	空				
試管 6	Yellow	Yellow	Yellow	Yellow	Yellow
試管 7	空				
試管 8	Black	Black	Black	Black	Black
試管 9	Green	Green	Green	Green	Green
試管 10	Red	Red	Red	Red	Red

+

【第三關】下面有9個填滿4種不同顏色的試管，以及2個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。(1★)



解題紀錄：

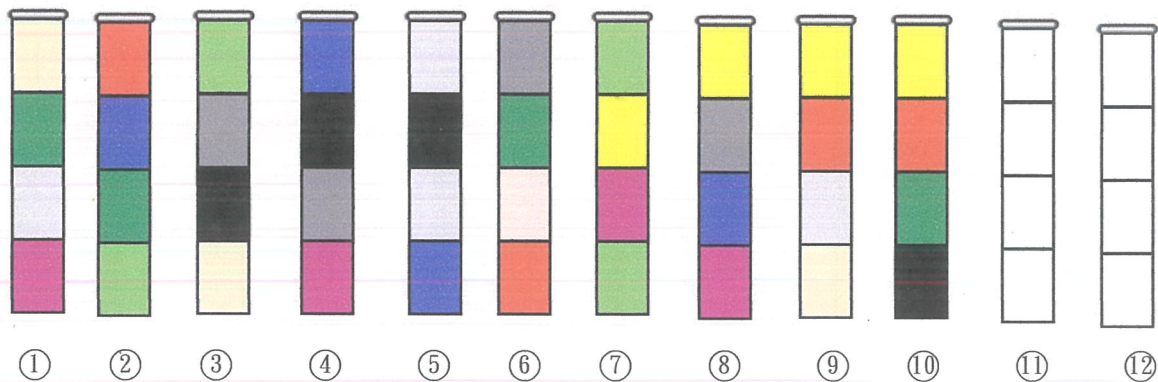
第 1-7 次	$\begin{matrix} 2 \\ (S) \end{matrix} \Rightarrow 10$	$\begin{matrix} 2 \\ (P) \end{matrix} \Rightarrow 11$	$\begin{matrix} 7 \\ (S) \end{matrix} \Rightarrow 10$	$\begin{matrix} 9 \\ (P) \end{matrix} \Rightarrow 11$	$\begin{matrix} 7 \\ (LtY) \end{matrix} \Rightarrow 9$	$\begin{matrix} 4 \\ (LtG) \end{matrix} \Rightarrow 7$	$\begin{matrix} 4 \\ (S) \end{matrix} \Rightarrow 10$
第 8-14 次	$\begin{matrix} 8 \\ (Gy) \end{matrix} \Rightarrow 4$	$\begin{matrix} 2 \\ (Y) \end{matrix} \Rightarrow 8$	$\begin{matrix} 2 \\ (S) \end{matrix} \Rightarrow 10$	$\begin{matrix} 8 \\ (Yx2) \end{matrix} \Rightarrow 2$	$\begin{matrix} 4 \\ (Gyx2) \end{matrix} \Rightarrow 8$	$\begin{matrix} 3 \\ (R) \end{matrix} \Rightarrow 4$	$\begin{matrix} 5 \\ (R) \end{matrix} \Rightarrow 4$
第 15-21 次	$\begin{matrix} 6 \\ (R) \end{matrix} \Rightarrow 4$	$\begin{matrix} 6 \\ (P) \end{matrix} \Rightarrow 11$	$\begin{matrix} 1 \\ (LtY) \end{matrix} \Rightarrow 5$	$\begin{matrix} 1 \\ (Y) \end{matrix} \Rightarrow 2$	$\begin{matrix} 7 \\ (LtGx2) \end{matrix} \Rightarrow 1$	$\begin{matrix} 6 \\ (G) \end{matrix} \Rightarrow 7$	$\begin{matrix} 6 \\ (P) \end{matrix} \Rightarrow 11$
第 22-28 次	$\begin{matrix} 3 \\ (B) \end{matrix} \Rightarrow 6$	$\begin{matrix} 3 \\ (Y) \end{matrix} \Rightarrow 2$	$\begin{matrix} 3 \\ (B) \end{matrix} \Rightarrow 6$	$\begin{matrix} 5 \\ (LtYx2) \end{matrix} \Rightarrow 3$	$\begin{matrix} 9 \\ (LtYx2) \end{matrix} \Rightarrow 3$	$\begin{matrix} 7 \\ (Gx2) \end{matrix} \Rightarrow 9$	$\begin{matrix} 5 \\ (LtG) \end{matrix} \Rightarrow 7$
第 29-35 次	$\begin{matrix} 9 \\ (Gx3) \end{matrix} \Rightarrow 5$	$\begin{matrix} 6 \\ (Bx2) \end{matrix} \Rightarrow 9$	$\begin{matrix} 8 \\ (Gyx3) \end{matrix} \Rightarrow 6$	$\begin{matrix} 8 \\ (B) \end{matrix} \Rightarrow 9$	$\begin{matrix} 1 \\ (LtGx3) \end{matrix} \Rightarrow 7$	$\begin{matrix} 6 \\ (Gyx3) \end{matrix} \Rightarrow 1$	$\Rightarrow$

最終結果：

試管 1	Gray	Gray	Gray	Gray	Gray
試管 2	Yellow	Yellow	Yellow	Yellow	Yellow
試管 3	LtYellow	LtYellow	LtYellow	LtYellow	LtYellow
試管 4	Red	Red	Red	Red	Red
試管 5	Green	Green	Green	Green	Green
試管 6	空				
試管 7	LtGreen	LtGreen	LtGreen	LtGreen	LtGreen
試管 8	空				
試管 9	Blue	Blue	Blue	Blue	Blue
試管 10	Silver	Silver	Silver	Silver	Silver
試管 11	Purple	Purple	Purple	Purple	Purple

+ 1

【第四關】下面有10個填滿4種不同顏色的試管，以及2個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。(2★)



解題紀錄：

第 1-7 次	8 → 11 (Y)	8 → 12 (Gy)	9 → 11 (Y)	10 → 11 (Y)	9 → 10 (R)	4 → 8 (B)	5 → 9 (S)
第 8-14 次	4 → 5 (Bk)	12 → 4 (Gy)	10 → 12 (Rx2)	2 → 12 (R)	2 → 8 (B)	2 → 10 (G)	3 → 2 (LtG)
第 15-21 次	3 → 4 (Gy)	5 → 3 (Bkx2)	5 → 9 (S)	8 → 5 (Bx3)	7 → 2 (LtG)	7 → 11 (Y)	8 → 7 (P)
第 22-28 次	9 → 8 (Sx3)	1 → 9 (LtY)	1 → 10 (G)	1 → 8 (S)	1 → 7 (P)	10 → 1 (Gx3)	3 → 10 (Bkx3)
第 29-35 次	9 → 3 (LtYx2)	4 → 9 (Gyx3)	6 → 9 (Gy)	6 → 1 (G)	6 → 3 (LtY)	12 → 6 (Rx3)	7 → 4 (Px3)

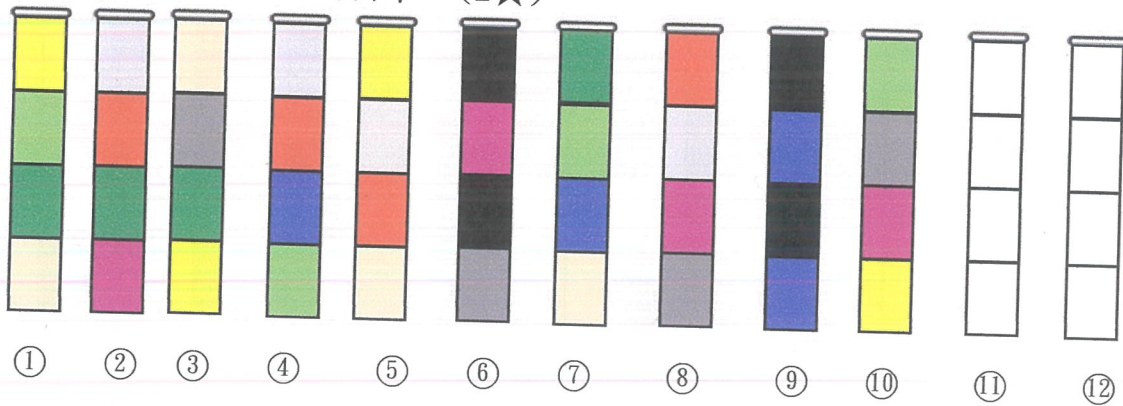
36. 2 → 7  
(LtGx3)

最終結果：

試管 1	Green	Green	Green	Green	
試管 2	空				
試管 3	LtYellow	LtYellow	LtYellow	LtYellow	
試管 4	Purple	Purple	Purple	Purple	
試管 5	Blue	Blue	Blue	Blue	
試管 6	Red	Red	Red	Red	
試管 7	LtGreen	LtGreen	LtGreen	LtGreen	
試管 8	Silver	Silver	Silver	Silver	
試管 9	Gray	Gray	Gray	Gray	
試管 10	Black	Black	Black	Black	
試管 11	Yellow	Yellow	Yellow	Yellow	
試管 12	空				

x 2

【第五關】下面有10個填滿4種不同顏色的試管，以及2個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。(2★)



解題紀錄：

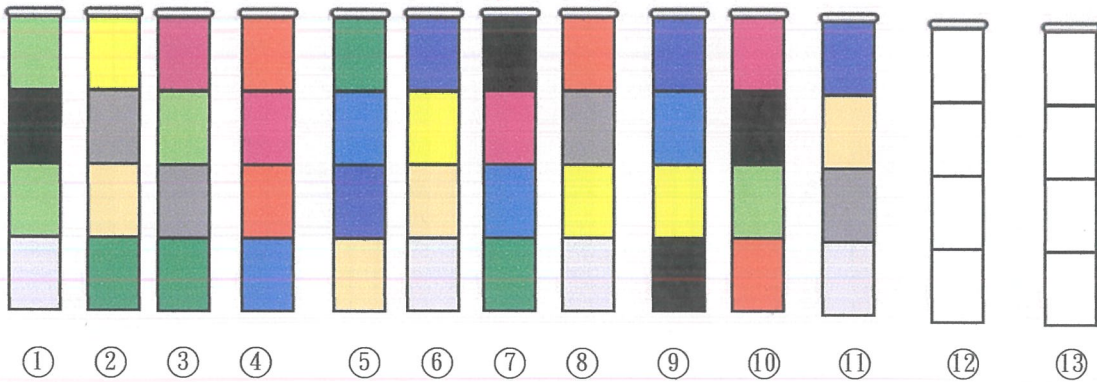
第 1-7 次	9 ⇒ 11 (Bk) ⇒	9 ⇒ 12 (B) ⇒	9 ⇒ 11 (Bk) ⇒	9 ⇒ 12 (B) ⇒	2 ⇒ 9 (S) ⇒	4 ⇒ 9 (S) ⇒	4 ⇒ 2 (R) ⇒
第 8-14 次	12 ⇒ 4 (Bx2) ⇒	1 ⇒ 12 (Y) ⇒	10 ⇒ 1 (LtG) ⇒	5 ⇒ 12 (Y) ⇒	5 ⇒ 9 (S) ⇒	6 ⇒ 11 (Bk) ⇒	8 ⇒ 5 (R) ⇒
第 15-21 次	8 ⇒ 9 (S) ⇒	6 ⇒ 8 (P) ⇒	6 ⇒ 11 (Bk) ⇒	10 ⇒ 6 (Gy) ⇒	10 ⇒ 8 (P) ⇒	10 ⇒ 12 (Y) ⇒	2 ⇒ 10 (Rx2) ⇒
第 22-28 次	5 ⇒ 10 (Rx2) ⇒	3 ⇒ 5 (LtY) ⇒	3 ⇒ 6 (Gy) ⇒	2 ⇒ 3 (G) ⇒	7 ⇒ 3 (G) ⇒	8 ⇒ 2 (Px3) ⇒	6 ⇒ 8 (Gy x3) ⇒
第 29-35 次	3 ⇒ 6 (Gx3) ⇒	12 ⇒ 3 (Yx3) ⇒	1 ⇒ 12 (LtGx2) ⇒	1 ⇒ 6 (G) ⇒	1 ⇒ 5 (LtY) ⇒	4 ⇒ 1 (Bx3) ⇒	12 ⇒ 4 (LtGx2) ⇒
第 36-38 次	7 ⇒ 4 (LtG) ⇒	7 ⇒ 1 (B) ⇒	5 ⇒ 7 (LtYx3) ⇒				

最終結果：

試管 1	Blue	Blue	Blue	Blue	Blue
試管 2	Purple	Purple	Purple	Purple	Purple
試管 3	Yellow	Yellow	Yellow	Yellow	Yellow
試管 4	LtGreen	LtGreen	LtGreen	LtGreen	LtGreen
試管 5	空	空	空	空	空
試管 6	Green	Green	Green	Green	Green
試管 7	LtYellow	LtYellow	LtYellow	LtYellow	LtYellow
試管 8	Gray	Gray	Gray	Gray	Gray
試管 9	Silver	Silver	Silver	Silver	Silver
試管 10	Red	Red	Red	Red	Red
試管 11	Black	Black	Black	Black	Black
試管 12	空	空	空	空	空

+ 2

【第六關】下面有11個填滿4種不同顏色的試管，以及2個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。(3★) 2



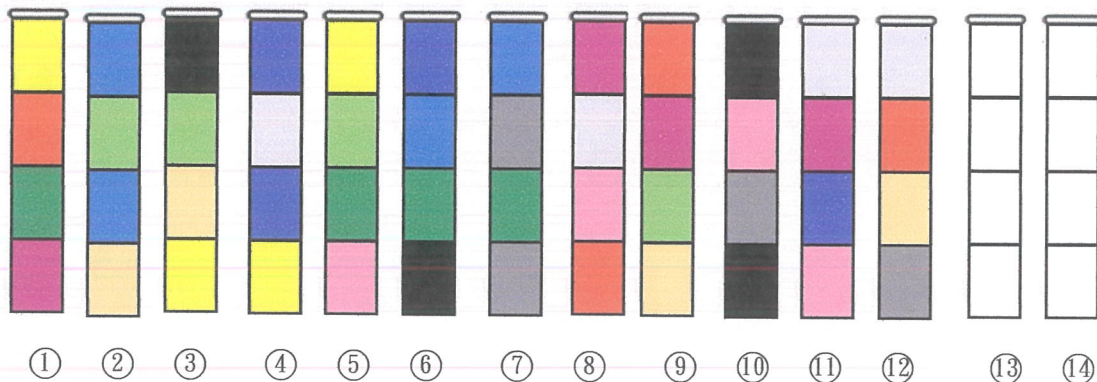
解題紀錄：

第 1-7 次	3 → 12 (P)	3 → 13 (LtG)	10 → 12 (P)	1 → 13 (LtG)	10 → 1 (Bk)	10 → 13 (LtG)	4 → 10 (R)
第 8-14 次	4 → 12 (P)	4 → 10 (R)	8 → 10 (R)	3 → 8 (Gy)	5 → 3 (G)	5 → 4 (LtB)	6 → 5 (B)
第 15-21 次	9 → 5 (B)	9 → 4 (LtB)	9 → 6 (Y)	7 → 9 (Bk)	7 → 12 (P)	7 → 4 (LtB)	3 → 7 (G)
第 22-28 次	5 → 3 (Bx3)	11 → 3 (B)	5 → 11 (LtY)	6 → 5 (Yx2)	11 → 6 (LtYx2)	8 → 11 (Gy x2)	5 → 8 (Yx2)
第 29-35 次	6 → 5 (LtYx3)	1 → 9 (Bk x2)	1 → 13 (LtG)	6 → 1 (S)	8 → 6 (Yx3)	8 → 1 (S)	11 → 8 (Gy x3)
第 36-42 次	11 → 1 (S)	2 → 6 (Y)	2 → 8 (Gy)	2 → 5 (LtY)	7 → 2 (Gx3)	→	→

最終結果：

試管 1	Silver	Silver	Silver	Silver	Silver
試管 2	Green	Green	Green	Green	Green
試管 3	Blue	Blue	Blue	Blue	Blue
試管 4	LtBlue	LtBlue	LtBlue	LtBlue	LtBlue
試管 5	LtYellow	LtYellow	LtYellow	LtYellow	LtYellow
試管 6	Yellow	Yellow	Yellow	Yellow	Yellow
試管 7	空				
試管 8	Gray	Gray	Gray	Gray	Gray
試管 9	Black	Black	Black	Black	Black
試管 10	Red	Red	Red	Red	Red
試管 11	空				
試管 12	Purple	Purple	Purple	Purple	Purple
試管 13	LtGreen	LtGreen	LtGreen	LtGreen	LtGreen

【第七關】下面有12個填滿4種不同顏色的試管，以及2個空試管，請利用這個空試管將四個試管倒成相同顏色的試管，在倒的過程中只能將相同顏色的顏料倒在一起而且不可以滿出來。(3★)



解題紀錄：

第 1-7 次	4 (B) → 13	4 (S) → 14	4 (B) → 13	5 (Y) → 4	6 (B) → 13	11 (S) → 14	8 (P) → 11
第 8-14 次	8 (S) → 14	12 (S) → 14	1 (Y) → 4	1 (R) → 12	2 (LtB) → 6	5 (LtG) → 2	1 (G) → 5
第 15-21 次	11 (Px2) → 1	11 (B) → 13	8 (Pk) → 11	9 (R) → 8	9 (P) → 1	12 (Rx2) → 8	2 (LtGx2) → 9
第 22-28 次	6 (LtBx2) → 2	5 (Gx2) → 6	5 (Pk) → 11	9 (LtGx3) → 5	9 (LtY) → 12	6 (Gx3) → 9	10 (Bk) → 6
第 29-35 次	10 (Pk) → 11	3 (Bk) → 6	3 (LtG) → 5	12 (LtYx2) → 3	10 (Gy) → 12	6 (Bkx3) → 10	2 (LtBx3) → 6
第 36-42 次	7 (LtB) → 6	7 (Gy) → 12	7 (G) → 9	7 (Gy) → 12	3 (LtYx3) → 2	4 (Yx3) → 3	→

最終結果：

試管 1	Purple	Purple	Purple	Purple	
試管 2	LtYellow	LtYellow	LtYellow	LtYellow	
試管 3	Yellow	Yellow	Yellow	Yellow	
試管 4	空				
試管 5	LtGreen	LtGreen	LtGreen	LtGreen	
試管 6	LtBlue	LtBlue	LtBlue	LtBlue	
試管 7	空				
試管 8	Red	Red	Red	Red	
試管 9	Green	Green	Green	Green	
試管 10	Black	Black	Black	Black	
試管 11	Pink	Pink	Pink	Pink	
試管 12	Gray	Gray	Gray	Gray	
試管 13	Blue	Blue	Blue	Blue	
試管 14	Silver	Silver	Silver	Silver	

+ 3



**【研究結果與發現】：必須超過50字(1★)**

1. 解題前先目測所有試管的顏色排列，先找出最佳的試管顏色開始倒。

何者是最佳顏色？

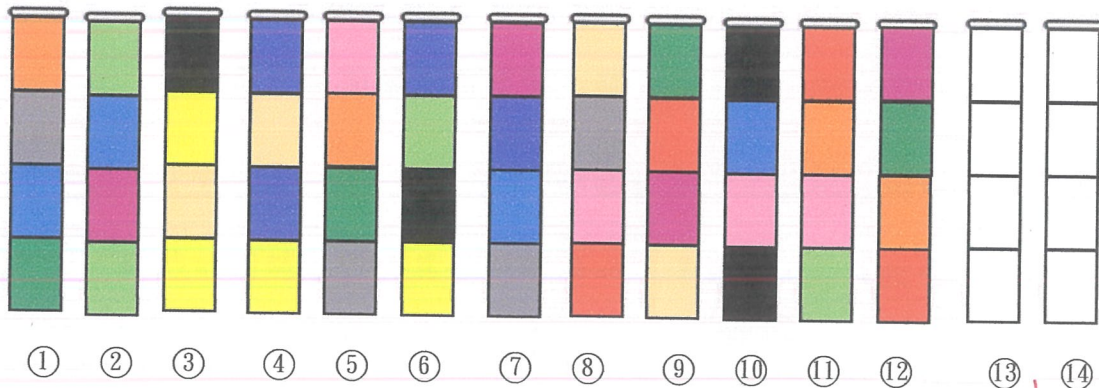
- (1) 在多個試管頂端有相同顏色, 如【第五關】中的黃色、灰色、黑色三種顏色。
  - (2) 最佳顏色上方的顏色數量要越少越好，因為可以比較容易倒出來，如【第五關】中的灰色、黑色，其中黃色因為在試管 3 及試管 10 上方共有 3 種顏色，所以不是最佳顏色。
2. 可以儘可能將某一支試管的水倒空，因為空的試管可以倒入的水空間最大，有利於其他試管的水倒入。
3. 當有空的試管產生時，空的試管要倒入顏色相同最多層的水, 可以容易達成最少的步驟完成倒水。
4. 相同顏色的水在同一試管中一定要一起倒完，如果目標沒有容量可以被一次倒完，就放棄不倒出跳過找其他試管需先處理其他。

依照上述心得執行會比較容易完成將水倒完組成相同顏色，最佳的試管顏色挑選會影響步驟的進行，不一定會是最少(最佳)步驟，但 3、4 的判斷執行會容易達到最少(最佳)步驟完成。

另詳見下方「補充說明」。

【題目設計】請利用12個填滿4種不同顏色的試管，以及2個空試管設計一道倒水關卡，並寫出解法。(2★)

設計如下圖：



解題紀錄：

第 1-7 次	7 → 13 (P)	7 → 14 (B)	12 → 13 (P)	9 → 12 (G)	11 → 9 (R)	1 → 11 (O)	4 → 14 (B)
第 8-14 次	8 → 4 (LtY)	8 → 1 (Gy)	5 → 8 (Pk)	6 → 14 (B)	2 → 6 (LtG)	7 → 2 (LtB)	1 → 7 (Gyx2)
第 15-21 次	2 → 1 (LtBx2)	13 → 2 (Px2)	11 → 13 (Ox2)	8 → 11 (Pkx2)	9 → 8 (Rx2)	5 → 13 (O)	12 → 5 (Gx3)
第 22-28 次	12 → 13 (O)	8 → 12 (Rx3)	9 → 8 (P)	2 → 8 (Px3)	4 → 9 (Ltyx2)	4 → 14 (B)	6 → 2 (LtGx2)
第 29-35 次	10 → 6 (Bk)	3 → 6 (Bk)	3 → 4 (Y)	3 → 9 (Lty)	3 → 4 (Y)	10 → 3 (LtB)	1 → 3 (LtBx3)
第 36-42 次	5 → 1 (Gx3)	7 → 5 (Gyx3)	10 → 7 (Pk)	11 → 7 (Pkx3)	11 → 2 (LtG)	6 → 10 (Bkx3)	4 → 6 (Yx3)

最終結果：

試管 1	Green	Green	Green	Green	Green
試管 2	LtGreen	LtGreen	LtGreen	LtGreen	LtGreen
試管 3	LtBlue	LtBlue	LtBlue	LtBlue	LtBlue
試管 4	空				
試管 5	Gray	Gray	Gray	Gray	Gray
試管 6	Yellow	Yellow	Yellow	Yellow	Yellow
試管 7	Pink	Pink	Pink	Pink	Pink
試管 8	Purple	Purple	Purple	Purple	Purple
試管 9	LtYellow	LtYellow	LtYellow	LtYellow	LtYellow
試管 10	Black	Black	Black	Black	Black
試管 11	空				
試管 12	Red	Red	Red	Red	Red
試管 13	Orange	Orange	Orange	Orange	Orange
試管 14	Blue	Blue	Blue	Blue	Blue

## 補充說明：

當第一次拿到本次高愛迪斯的題目時，回家有空時測試了【第一關】及【第七關】，【第一關】很簡單，但【第七關】非常難，測試了很久很久才找到解答，共用了步驟 50 次，查詢了一下 Google 大神，發現有類似的網路遊戲「水排序」，遊戲的同時也對於其如何解題而不得其解，漸漸摸索才漸漸有了上述【研究結果與發現】的心得，且同樣的顏色組合完成步驟卻有不同，直到最後完成時，步驟可能多也可能少，**解法步驟不只一種。**

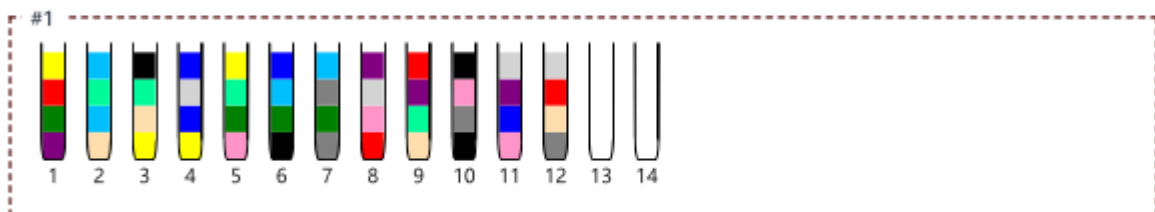
網上有一個網址(見下方參考資料：水排序求解

<http://www.elfin.cc/water-sort-solver>)，試著嘗試當輸入了跟本次高愛迪斯的作業

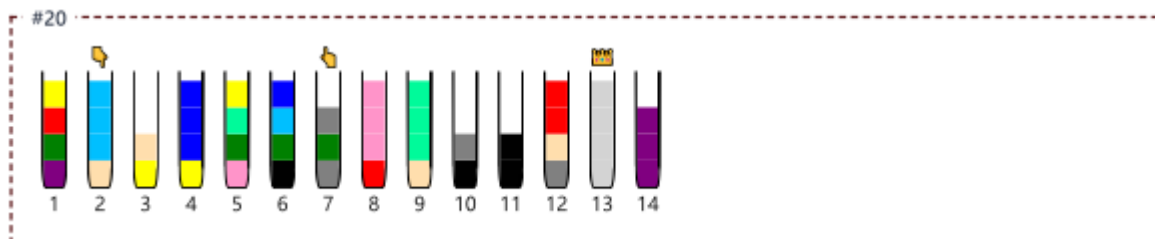
相同的【第七關】相同顏色參數時，得到總共完成步驟數是 46 個步驟：

網上的解答(如下圖)：

答案

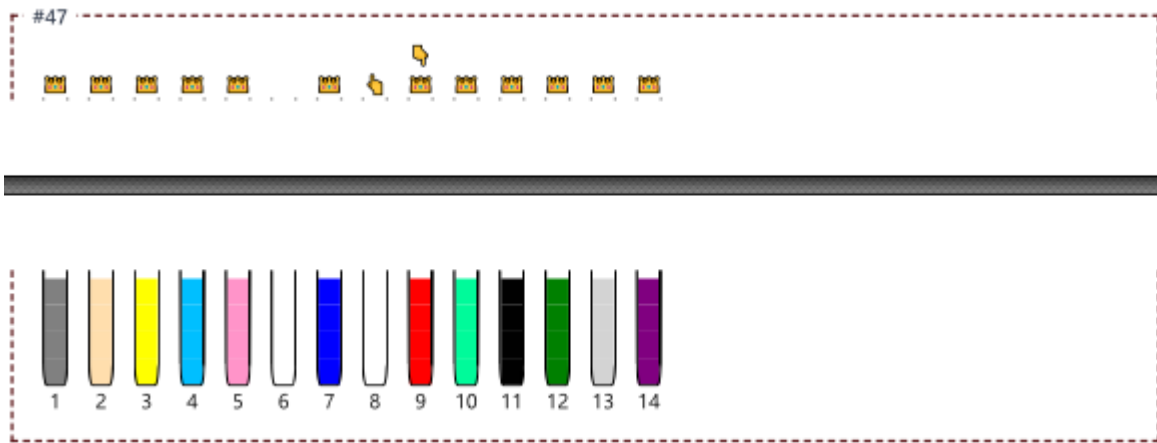


...(省略)



...(省略)

最後：



總共 46 步驟(47-1)。

顯然跟自己的 50 步有一些差異，於是找了相關資料與研究才有了上述的【**研究結果與發現**】經由此發現，再鍥而不捨修正可以進步到 48 次。

在研究過程中請教爸爸時，發現本次高愛迪斯的題目時，類似一個數學問題河內之塔 (Tower of Hanoi) (見下方參考資料)：這是根據一個傳說形成的數學問題，最早發明這個問題的人是法國數學家愛德華·盧卡斯(參考維基百科 [https://en.wikipedia.org/wiki/Tower\\_of\\_Hanoi](https://en.wikipedia.org/wiki/Tower_of_Hanoi))。由於完成河內之塔的步驟次數是塔高=N，最少需移動  $2^n-1$  次，這是一個人類搬不完的工作。如果用電腦來搬，解法也是用遞迴(recursive) (參考同上維基百科)。此工作跟排列組合或搜尋一樣，都可以用遞迴原理來解。

因遞迴是一個重覆執行自己的迴圈，程式很簡潔，主要是依靠堆疊返回再重覆執行自己的執行演算法，因此遞迴的缺點是執行次數非常多，就像數學排序或搜尋一樣，需要有最佳演算法來縮短解決時間，不然執行效率會很差，例如當試管數量多(如 12 支試管或更多時)執行起來會很耗時，就必需要有一些最佳化判斷條件如上述的【**研究結果與發現**】，由於爸爸會寫程式，請他幫忙依上述的【**研究結果與發現**】設計了一個程式(使用 Java)來驗證，為了驗證最佳

顏色選擇，程式設計成可以輸入參數從第幾個試管開始倒，且因程式執行是機器不是人，除上述的【研究結果與發現】條件外，還需要有一些遞迴停止的條件如：14 支試管，每支試管 4 種顏色，最多遞迴深度設為  $14 \times 4 = 56$  次，當達到最大遞迴深度時就需返回，另當各別試管的顏色都已相同時，表示該試管已完成就不要處理該試管的倒出動作，直到完成全部試管為止。

最後用【第七關】驗證程式，設定從第 12 支試管開始倒得到完成步驟次數=49，從第 4 支試管開始倒，得到完成步驟次數=43，將程式設定從第 1 支試管、第 2 支試管..到第 12 支試管分別開始倒水，跑完的結果如下：

### 【第七關】

啟始試管	完成次數
1	無法完成
2	50
3	50
4	43
5	無法完成
6	48
7	48
8	48
9	49
10	46
11	46
12	49

如果再加入【研究結果與發現】中的發現：當有空的試管出現時，**空的試管要倒入顏色相同最多層的水**，如果從第 4 支試管開始倒，得到完成步驟次數=41，有最佳化的結果，這個結果比網路上「水排序求解」的完成步驟 46 **還最佳且少了 5 個步驟**。

啟始試管	完成次數
4	41

最後將其他【第一關】到【第六關】及【題目設計】的執行完成結果列表如下：

【第一關】

【第二關】

【第三關】

【第四關】

啟始試管	完成次數	啟始試管	完成次數	啟始試管	完成次數	啟始試管	完成次數
1	21	1	29	1	40	1	42
2	19	2	31	2	34	2	38
3	21	3	29	3	35	3	41
4	19	4	30	4	34	4	44
5	20	5	34	5	35	5	41
6	20	6	30	6	35	6	37
		7	33	7	34	7	37
		8	32	8	36	8	36
				9	41	9	38
						10	38

【第五關】

【第六關】

【題目設計】

啟始試管	完成次數
1	38
2	38
3	39
4	41
5	40
6	42
7	無法完成
8	40
9	38
10	40

啟始試管	完成次數
1	46
2	45
3	41
4	43
5	43
6	43
7	42
8	43
9	41
10	43
11	42

啟始試管	完成次數
1	45
2	43
3	47
4	48
5	46
6	43
7	42
8	47
9	47
10	48
11	47
12	47

由於本次作業不是程式的研究，請爸爸寫的驗證程式尚有改進空間，且只是

做為輔助驗證，僅列表在最後附件供參考。

程式的執行

java TubesX 12 2 12.txt 4

參數 1：12 是指共有 12 支試管

參數 2：2 是指共有 2 支試空試管

參數 3：12.txt 是顏色的定義檔

參數 4：4 是第 4 支試管開始倒

其中 12.txt 顏色的定義檔內容：(註：顏色可以輸入中文,如：黃,紅,綠,紫..等)

Yellow, Red, Green, Purple  
LtBlue, LtGreen, LtBlue, LtYellow  
Black, LtGreen, LtYellow, Yellow  
Blue, Silver, Blue, Yellow  
Yellow, LtGreen, Green, Pink  
Blue, LtBlue, Green, Black  
LtBlue, Gray, Green, Gray  
Purple, Silver, Pink, Red  
Red, Purple, LtGreen, LtYellow  
Black, Pink, Gray, Black  
Silver, Purple, Blue, Pink  
Silver, Red, LtYellow, Gray

參考資料：

河內之塔：[https://en.wikipedia.org/wiki/Tower\\_of\\_Hanoi](https://en.wikipedia.org/wiki/Tower_of_Hanoi)

水排序求解：<http://www.elfin.cc/water-sort-solver>

水排序謎題：<https://blog.csdn.net/nameofcsdn/article/details/117620445>

遞迴：<https://en.wikipedia.org/wiki/Recursion>

深度優先搜尋：[https://en.wikipedia.org/wiki/Depth-first\\_search](https://en.wikipedia.org/wiki/Depth-first_search)

附件：

程式列表：

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class TubesX { //TubeColorProblem {
    static int count = 0;
    static long try_count = 0;
    static long back_count = 0;
    static int tubeColorCnt = 1;
    static int tubeEmptyCnt = 1;
    static int tubeTotalCnt; // 總共 tube 數量(含空 tube)
    static int startTube = 1; // 從第(1~ n)個 tube 開始倒
    static int maxSteps = 56; // 定義最多 56 步 (14 tube* 4colors)
    static long t1;
    static long t2;
    static int verbose; // 印出搬動過程,1:印過程,2:印過程+內容
    static ArrayList<String []> stepx = new ArrayList<String []>(); // 記錄最後搬動步驟
    static ArrayList<String []> tubeColorProp = new ArrayList<String []>(); // 讀入 tubes 顏色資訊內容
    static int layers = 4; // 每個 tube 共 4 層
    static List<List<String>> tubes;

    private static void printAllTubes(int level)
    {
        if (level >= 3){
            for (int k = 0; k < tubes.size(); k++) {
                System.out.println("Tube " + (k + 1) + ": " + tubes.get(k));
            }
        }
    }

    private static String getDateTime() {
        int mils = (int)(System.currentTimeMillis() % 1000) ;
        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss.");
        Date date = new Date();
        String timeSec= dateFormat.format(date);
        return timeSec+mils;
    }

    private static String getElapsedTime(long tx) {
        long millis = tx % 1000;
        long second = (tx / 1000) % 60;
        long minute = (tx / (1000 * 60)) % 60;
        long hour = (tx / (1000 * 60 * 60)) % 24;
        long day = (tx / (1000 * 60 * 60 * 24)) ;
        String time = String.format("%02d-%02d-%02d:%02d:%d", day, hour, minute, second, millis);
        return time;
    }

    private static int getSameColorsCnt(int n) { // 計算 tube 中,上層相同顏色的數量
        int size = tubes.get(n).size();
        if (size == 0)
            return 0;
        String color = tubes.get(n).get(0);
```



```

int cnt = 1;
for (int j = 1; j < size; j++) {
    if (color.equalsIgnoreCase(tubes.get(n).get(j)) )
        cnt++;
    else
        break;
}

return cnt;
}

private static boolean isSameColors(int n) { // tube 中,是否全部是相同顏色
int size = tubes.get(n).size();
if (size < layers)
    return false;
for (int j = 0; j < (layers-1); j++) { // 檢查每個 tube 4 層顏色是否相同,只需檢查 3 層(layers-1)即可
    if (!tubes.get(n).get(j).equalsIgnoreCase(tubes.get(n).get(j+1)) )
        return false;
    }
return true;
}

private static boolean canPour(int from, int to) {
if (tubes.get(from).isEmpty() || from == to)
    return false; // 來源是空的 或 目標同一個 tube->不可倒 (註:同一目標在前述中已判斷)
if (isSameColors(from))
    return false; // 同一顏色已滿的 tube,不要搬走
int size_to = tubes.get(to).size();
int size_from = tubes.get(from).size();
if (size_to >= layers) // tube 顏色已滿,不可倒
    return false;
int from_same_cnt = getSameColorsCnt(from);
if (size_to == 0 && from_same_cnt == size_from)
    return false; // 空的 tube->可倒 ->但要排除來自同一顏色且未滿的 tube (**** 本判斷非常重要 ****)
if (size_to == 0)
    return true; // 空的 tube->可倒
String color_fm = tubes.get(from).get(0); // 取 from 最上層顏色
String color_to = tubes.get(to).get(0); // 取 to 最上層顏色
if (!color_fm.equalsIgnoreCase(color_to))
    return false; // 顏色不相同->不可倒
if ( from_same_cnt > 1 && (from_same_cnt+size_to) > layers) // 同一顏色的 tube 無法一次倒->不可倒,效率會好一點,
// 步驟會少一點 (**** 本判斷也重要 ****)
    return false;
return true;
}

private static int pourMove(int from, int to) {
if (!tubes.get(from).isEmpty()) {
    String from_c = tubes.get(from).get(0);
    int cnt = 0;
    while (tubes.get(from).size() > 0 && from_c.equalsIgnoreCase(tubes.get(from).get(0)) && tubes.get(to).size() < layers)
// 將相同顏色的層一次全搬走,例如要搬紅色,但紅色有 2 層->一次全部搬走
    {
        tubes.get(to).add(0,tubes.get(from).remove(0));
        cnt++;
    }
    return cnt;
}
return 0;
}
}

```

```

private static void pourBack(int from, int to, int cnt) {
    if (cnt == 0)
        return;
    if (!tubes.get(from).isEmpty()) {
        for (int i = 0; i < cnt; i++)
            tubes.get(to).add(0, tubes.get(from).remove(0));
    }
}

static boolean isAllSameColors() { // 檢查是否所有的 Tube 都是相同顏色 或空的, recursive 結束
    for (int i = 0; i < tubes.size() - 1; i++) {
        if (tubes.get(i).size() == 0)
            continue;
        if (tubes.get(i).size() < layers) // tube 尚未有 4 層顏色 (layers=4)
            return false;
        for (int j = 0; j < (layers-1); j++) { // 檢查每個 tube 4 層顏色是否相同, 只需檢查 3 層(layers-1)即可
            if (!tubes.get(i).get(j).equalsIgnoreCase(tubes.get(i).get(j+1)))
                return false;
        }
    }
    return true;
}

static boolean WaterSolve(int steps) {
    if (steps > maxSteps || isAllSameColors()) // 如果超過步驟或已完成->返回完成
        return true;
    int i; // 設定從第 i tube 開始倒
    for (int i2 = 0; i2 < tubeTotalCnt; i2++) {
        i = startTube+i2-1; // 第 i tube 開始倒
        if (i >= tubeTotalCnt) i -= tubeTotalCnt;
        if (steps == 0 && i2 != 0) // 如果回到 step=0->表示無法搬
            return false;
        for (int j = 0; j < tubes.size(); j++) {
            try_count++;
            int movecnt = 0;
            if (try_count % 2100000000 == 0) // 只是在過程太久中輸出 console, 知道程式還在執行
            {
                long t = System.currentTimeMillis() % 2;
                if (t == 0)
                    System.out.print(".");
                else
                    System.out.print("*");
            }
            if (i != j) {
                if (canPour(i, j)) // 判斷是否可倒->處理 倒
                {
                    String c = tubes.get(i).get(0);
                    movecnt = pourMove(i, j);
                    t2 = System.currentTimeMillis();
                    long tx = t2-t1;
                    count++;
                    if (verbose >= 2)
                        System.out.println("Pour from Tube[WaterSolve]:第"+count+"次 [step="+steps+"]: "+ (i + 1) + " to
Tube " + (j + 1) + ", 搬 "+movecnt+" 個", try_count="+try_count+" ["+getDate()+"], ["+getElapsedTime(tx)+"]);
                    String [] stx = new String[4];
                    stx[0]= String.valueOf(i+1);stx[1]=String.valueOf(j+1);stx[2]=String.valueOf(movecnt);stx[3]=c;
                    stepx.add(stx); // 將步驟存起來, 含顏色
                    printAllTubes(verbose); // 印出所有 Tubes
                    if (WaterSolve(steps + 1))

```

```

        return true;
        pourBack(j, i, movecnt); // 無法完成->搬回去
        count--;
        back_count++;
        stepx.remove(steps);
        if (verbose >= 2)
            System.out.println("Back+++ from Tube[WaterSolve]:第"+count+"次 [step="+steps+"]: "+(j+1)+"
to Tube "+(i+1)+",搬 "+movecnt+" 個");
        //if (steps == 0 && i != (startTube-1)) // 如果回到 step=0->表示無法搬
        // return false;
        } // if canPour()
    }
} // for (j..)
} // for (i..)
return false;
}

static void tubeInit(String infile) {
    FileReader fr = null;
    BufferedReader br = null;
    String str;
    String[] items;

    tubeColorProp.clear();
    try {
        fr = new FileReader(infile);
        br = new BufferedReader(fr);
        while ((str = br.readLine()) != null) {
            items = str.trim().split(",");
            tubeColorProp.add(items);
        }
    }
    catch (Exception e){
        System.out.println();
        System.out.println("ERROR f00: " + e.getMessage());
        e.printStackTrace();
    }
}

}

public static void main(String[] args) {
    System.out.println("開始: "+getDateTime());
    t1 = System.currentTimeMillis();
    tubeColorCnt = 6;
    tubeEmptyCnt = 1;
    startTube = 1; // 預設定從第 1 個 tube 開始倒
    if (args.length < 3) // 讀入 Tubes 資訊 .txt
    {
        System.out.println("參數錯誤,至少輸入 3 參數[參數 1:tube 數量],[參數 2:空 tube 數量],[參數 3: 顏色檔案名
稱],[參數 4: 從第幾個 tube 開始搬]");
        System.exit(0);
    }
    tubeColorCnt = Integer.parseInt(args[0]);
    tubeEmptyCnt = Integer.parseInt(args[1]);
    tubeInit(args[2]);
    if (args.length >= 4) // 設定起始 tube
    {
        int n = Integer.parseInt(args[3]);
        if (n >= tubeColorCnt)
            n = tubeColorCnt;
        startTube = n;
    }
}

```

程

```
verbose = 0;
if (args.length >= 5) // 是否輸出過程 1:過程不輸出,最後輸出已完成的過程, 2:只輸出簡易過程, 3:輸出每一步過
程
    verbose = Integer.parseInt(args[4]);
System.out.println("tubeColorCnt=" +tubeColorCnt+", tubeEmptyCnt="+tubeEmptyCnt);
tubeTotalCnt = tubeColorCnt + tubeEmptyCnt;
System.out.println("從 tube "+startTube+" 開始,共有 "+tubeTotalCnt+" tubes");

tubes = new ArrayList<>();
for (int i = 0; i < tubeTotalCnt; i++) { // 新增總共的 tubes 數(含空的 tube)
    tubes.add(new ArrayList<>());
}
for (int i = 0 ; i < tubeColorProp.size() ; i++)// 將讀進顏色設定檔的顏色資訊->放入 Tube 中
{
    int len = tubeColorProp.get(i).length;
    for (int j = 0 ; j < len ; j++)
        tubes.get(i).add(tubeColorProp.get(i)[j].trim());
}
printAllTubes(3);
if (WaterSolve(0)) { // 執行找出解答 Recursive ->輸入有有顏色的 tube 數量
    // 已找到完成
    System.out.println("");
    for (int i = 0; i < stepx.size(); i++) { // 印出所有步驟
        String[] st = stepx.get(i);
        System.out.println("Step "+ (i+1) +" From " + st[0] + " to " + st[1]+" 搬 "+st[2]+" 個 "+st[3]);
    }
    if (verbose > 0 && verbose <= 2) { // 是否設定要印出最後完成過程的 Tubes 變化過程
        tubeInit(args[2]); // 重讀 Tubes 顏色
        for (int i = 0 ; i < tubes.size() ; i++) // 清 tubes 中的顏色值
            tubes.get(i).clear();;
        // 印出完成的步驟
        for (int i = 0 ; i < tubeColorProp.size() ; i++){
            int len = tubeColorProp.get(i).length;
            for (int j = 0 ; j < len ; j++)
                tubes.get(i).add(tubeColorProp.get(i)[j].trim());
        }
        // 印出 完整 tubes 變化過程
        System.out.println("+++++++ 完整過程 ++++++++");
        for (int i = 0; i < stepx.size(); i++) {
            String[] st = stepx.get(i);
            System.out.println("+++++ Step "+ (i+1) +" +++++ From " + st[0] + " to " + st[1]+" 搬 "+st[2]+" 個 "+st[3]);
            pourMove(Integer.valueOf(st[0])-1,Integer.valueOf(st[1])-1);
            printAllTubes(3);
        }
    }
    System.out.println("\n 完成: try_count="+try_count+", back_count="+back_count);
    t2 = System.currentTimeMillis();
    long tx = t2-t1;
    System.out.println(getDateTime()+", 共費時:"+getElapsedTime(tx)+",起始第 "+startTube+" tube,次數="+count);
    for (int i = 0; i < tubes.size(); i++) {
        System.out.println("Tube " + (i + 1) + ": " + tubes.get(i));
    }
} else { // 無法完成
    System.out.println("\n 無法完成: try_count="+try_count+", back_count="+back_count);
    t2 = System.currentTimeMillis();
    long tx = t2-t1;
    System.out.println(getDateTime()+", 共費時:"+getElapsedTime(tx)+",起始第 "+startTube+" tube,次數="+count);
```

```
        for (int i = 0; i < tubes.size(); i++) {  
            System.out.println("Tube " + (i + 1) + ": " + tubes.get(i));  
        }  
    }  
}
```