

#83

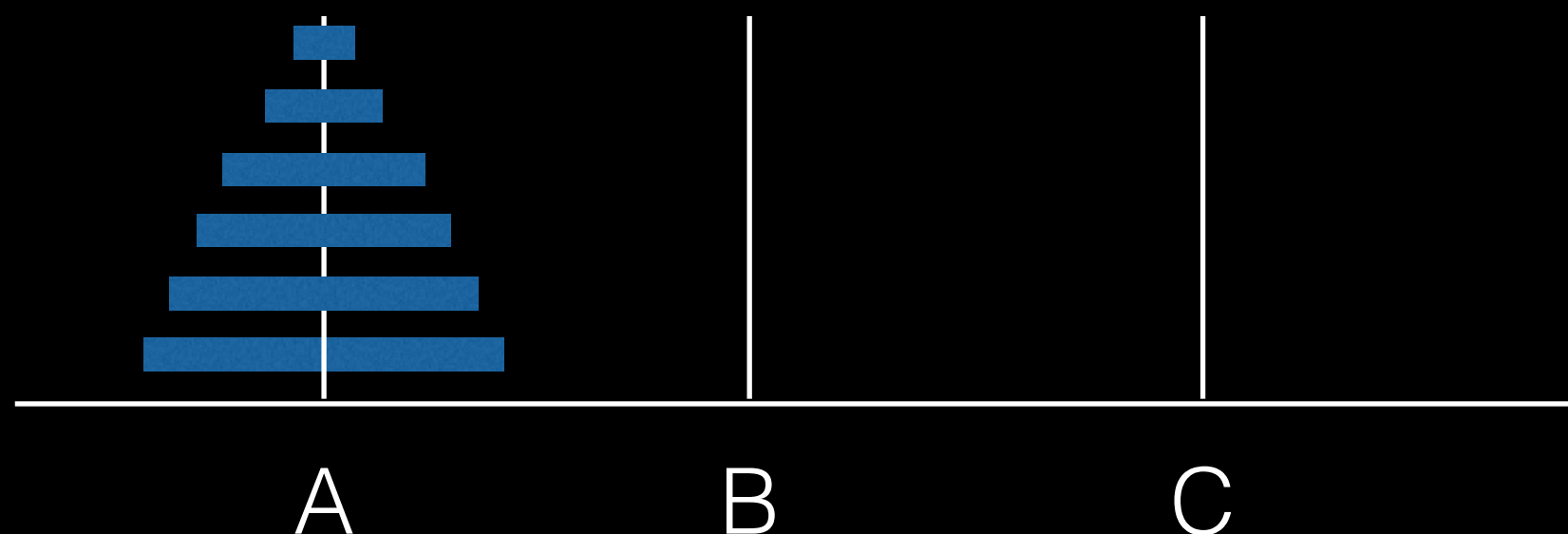
Restricted Tower of Hanoi

NickWarm

a.k.a

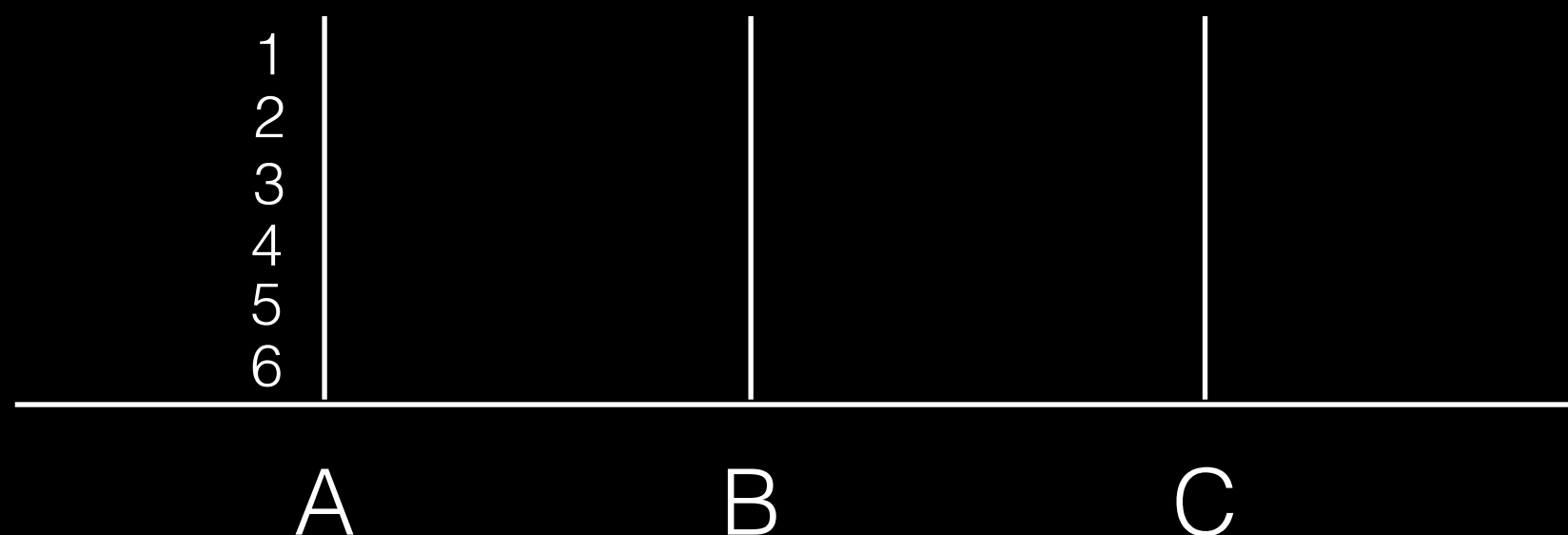
WG

原始的河內塔



目標：把碟子移動到C

簡化問題

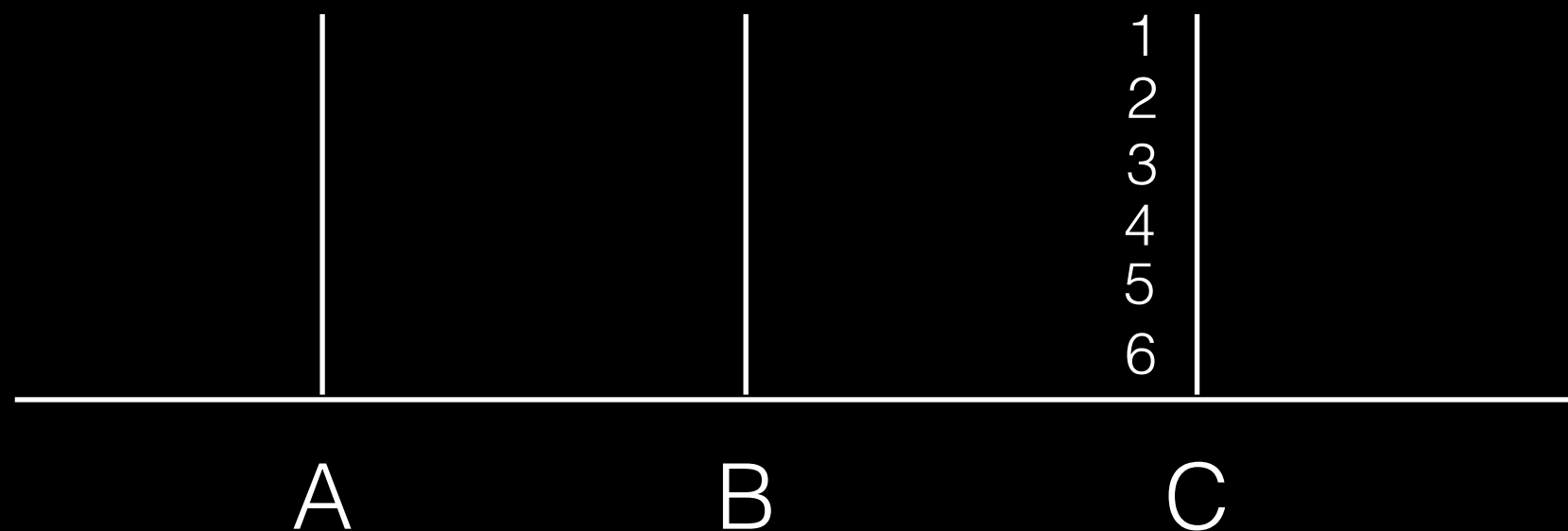


用數字來代表盤子大小
 $n = 6$

條件

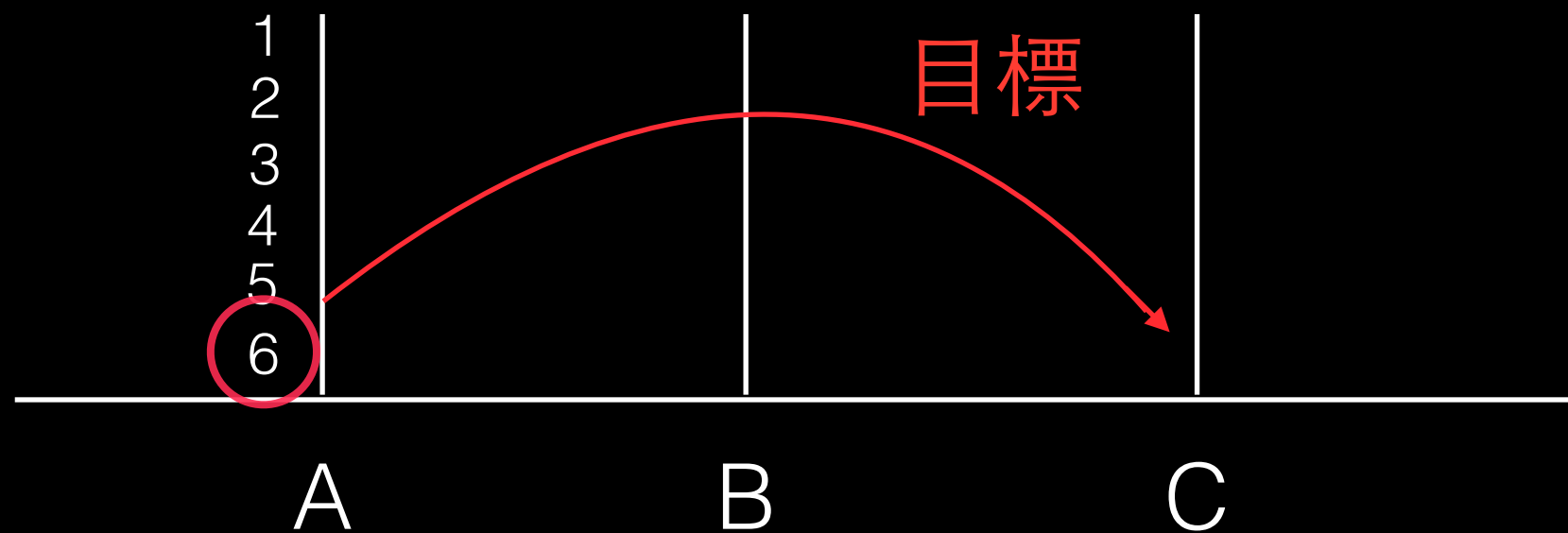
- 一次移動一個碟子
- 大的碟子不能疊在小的上面

期待の結果

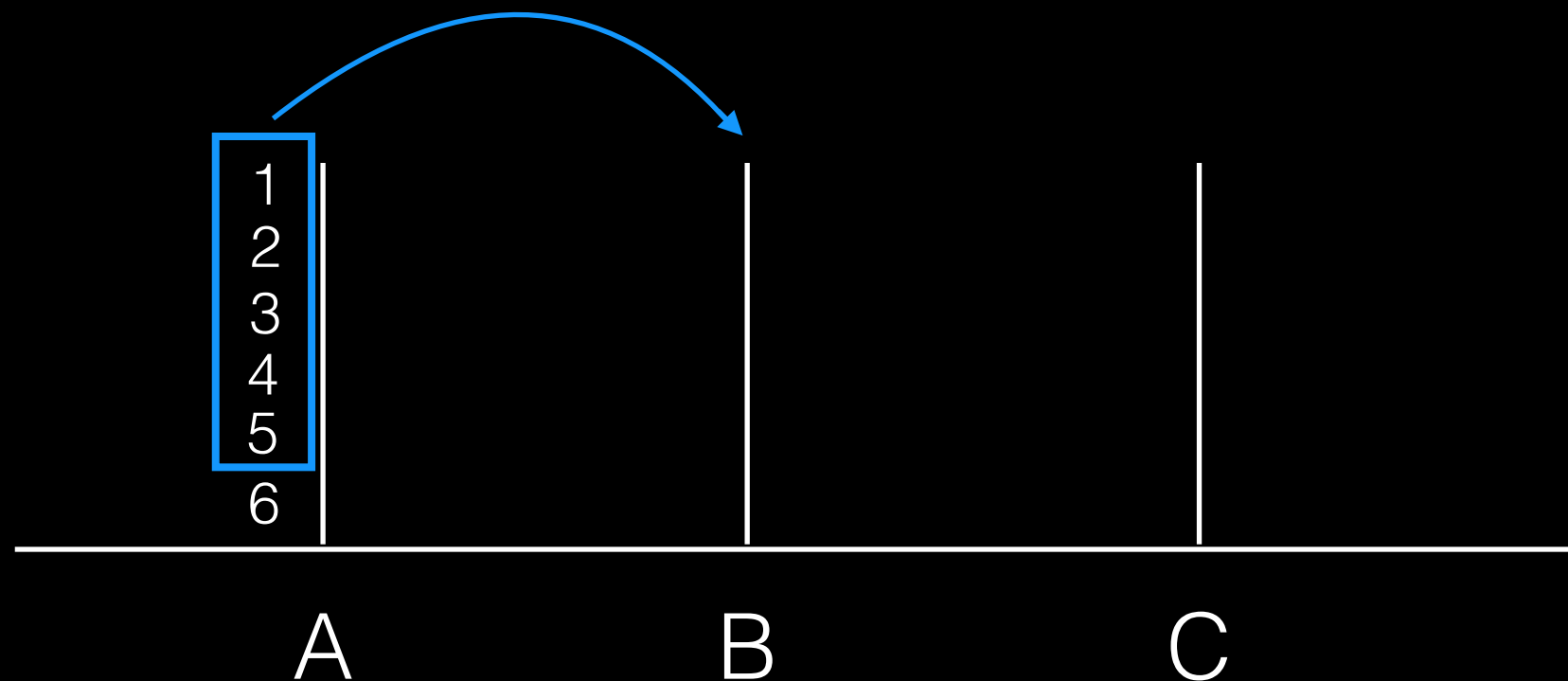


$$n = 6$$

目標：最大的碟子，要先放在柱子C

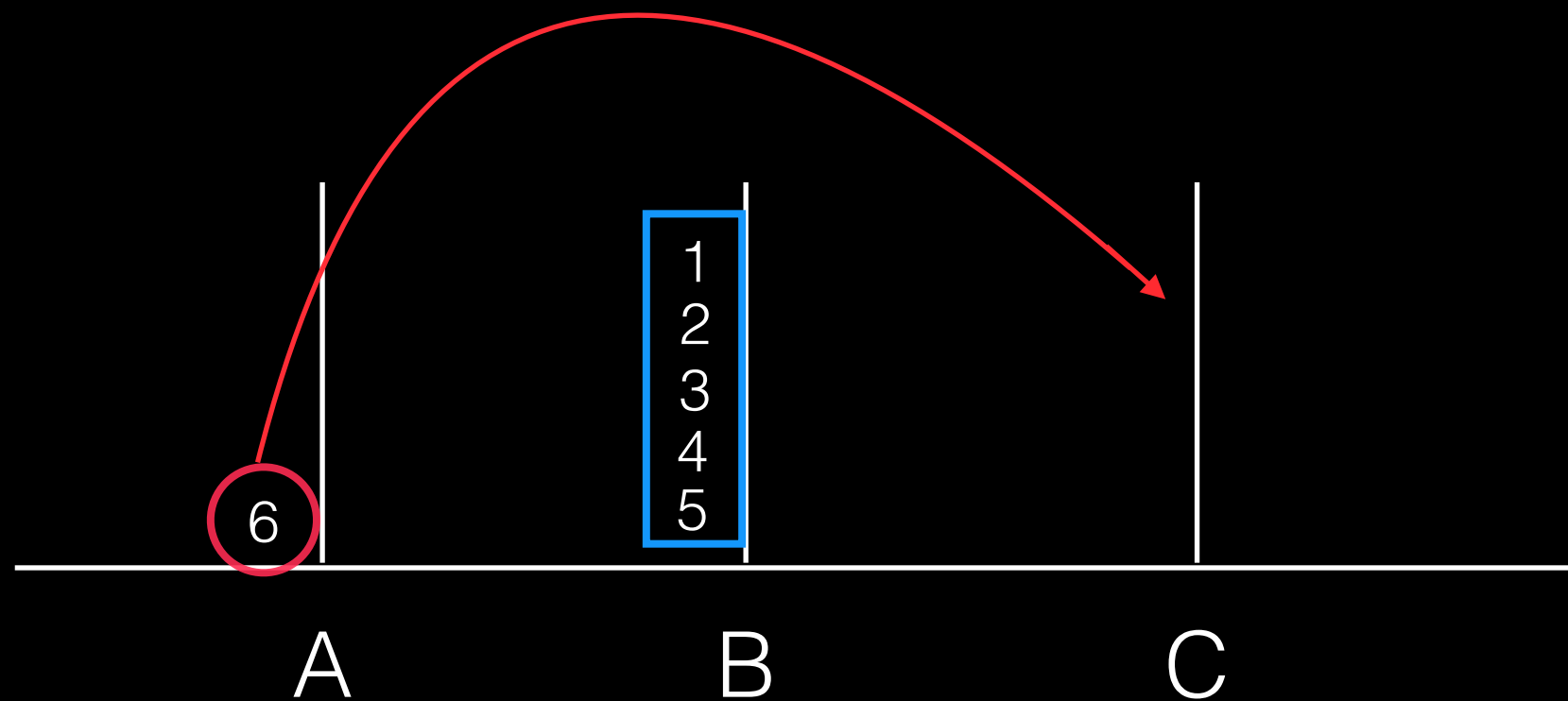


所以：上面的碟子，要先放在柱子 B



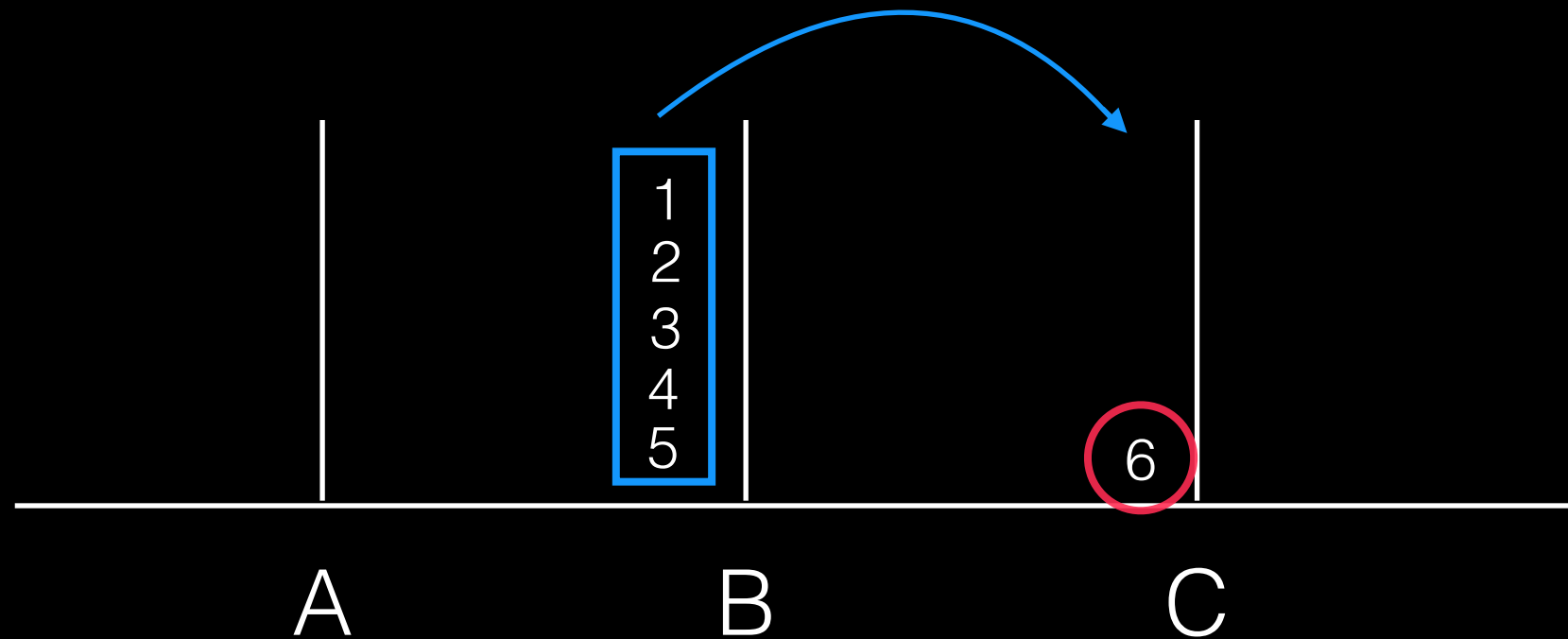
先把 $n-1$ 個碟子從A放到B
 $n = 6$

然後：最大的碟子，就能放在柱子C



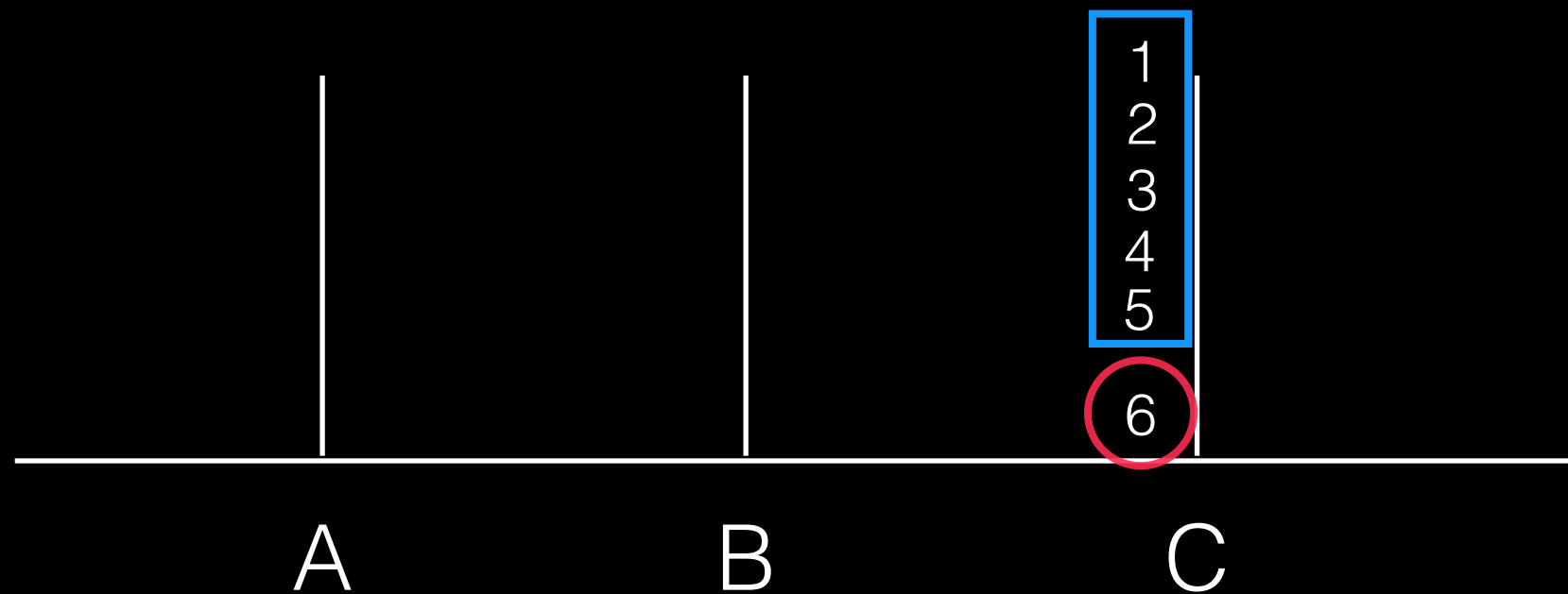
把1個碟子從A放到C
 $n = 6$

最後：把剩下的碟子放到柱子C



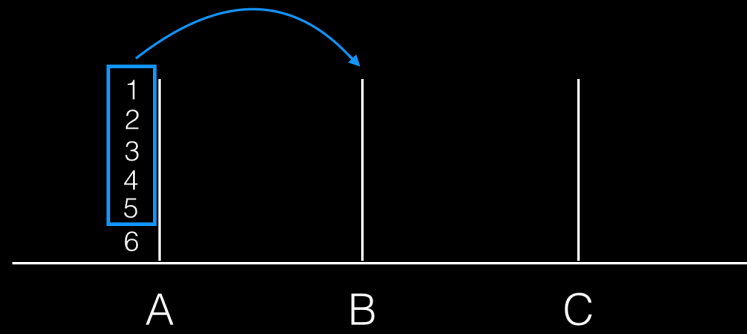
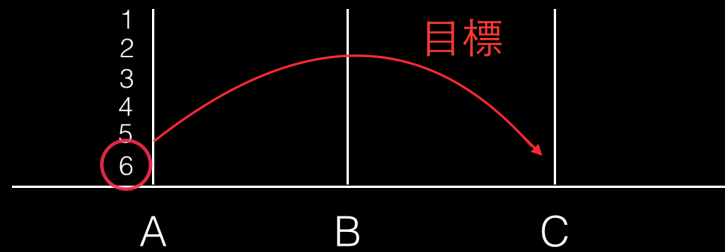
先把 $n-1$ 個碟子從B放到C
 $n = 6$

完成

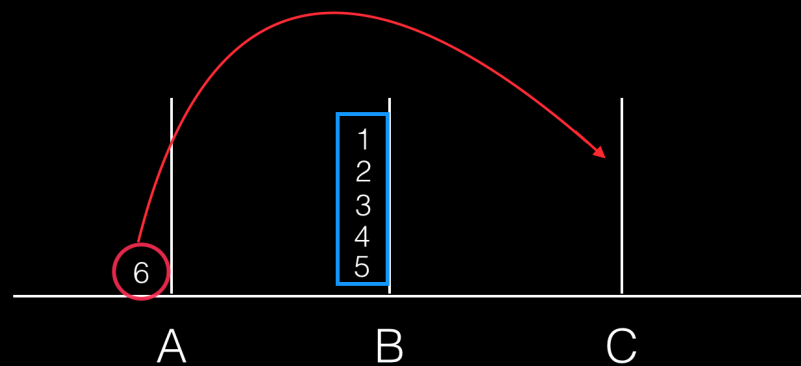


$n = 6$

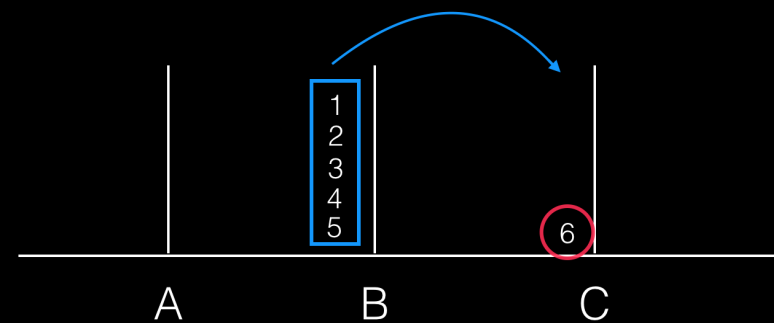
整個流程



n-1個碟子從A移到B



1個碟子從A移到C



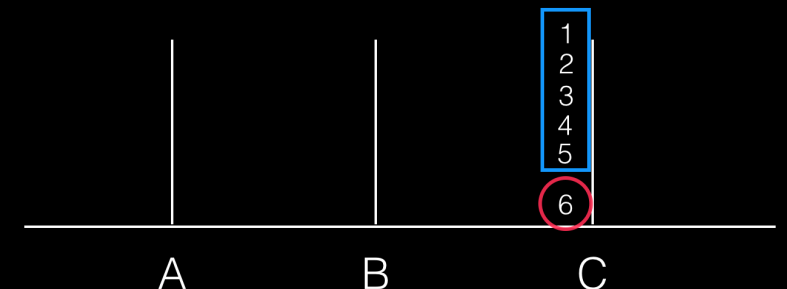
n-1個碟子從B移到C

河內塔的演算法

$$T(n) = \boxed{2}T(n-1) + \boxed{1}$$

↓
n-1個
碟子
移動兩次
1個碟子
移動一次

就會變得
非常直覺



程式碼

- demo

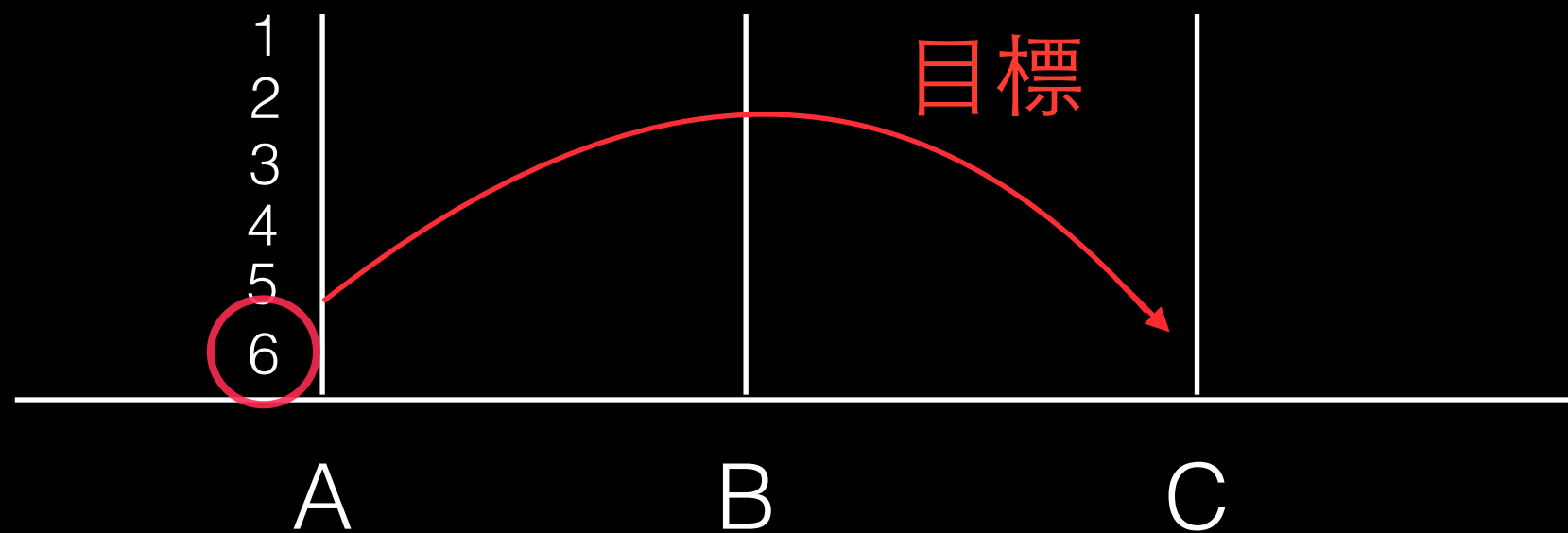
#83

Restricted Tower of Hanoi

條件

- 一次移動一個碟子
- 大的碟子不能疊在小的上面
- 一定要通過中間的柱子B

目標：最大的碟子，要先放在柱子C



依照相同的概念

BUT !!!

條件

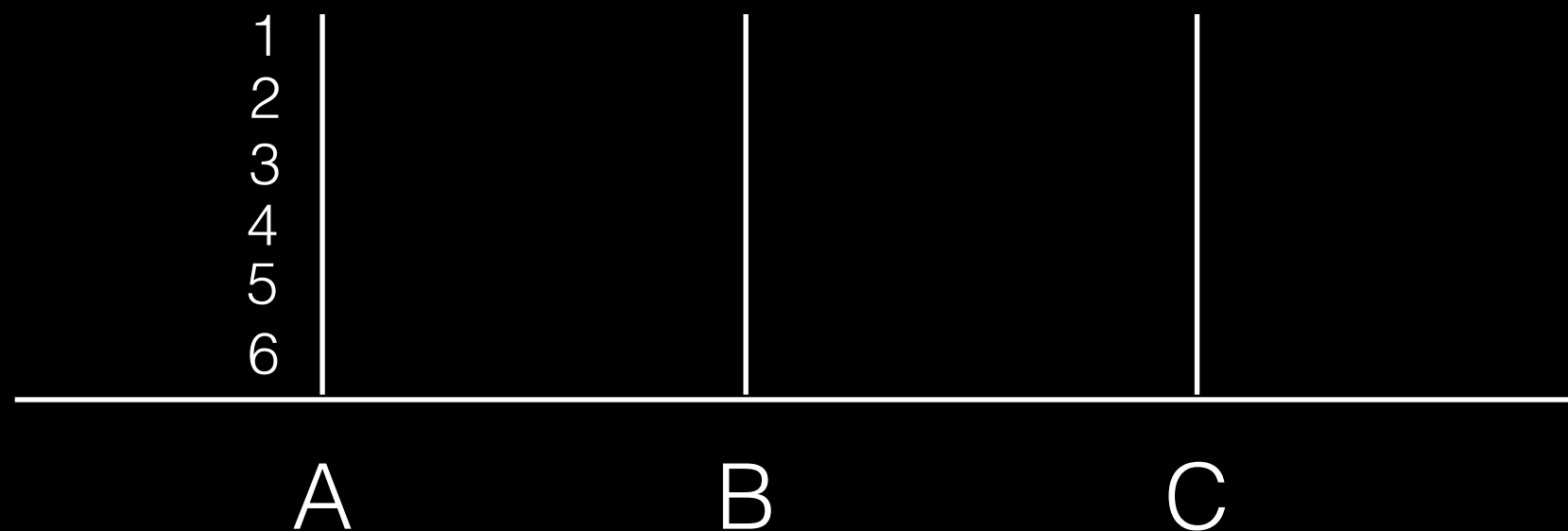
- 一次移動一個碟子
- 大的碟子不能疊在小的上面
- 一定要通過中間的柱子B

SO,

最大的碟子要先到達B柱子

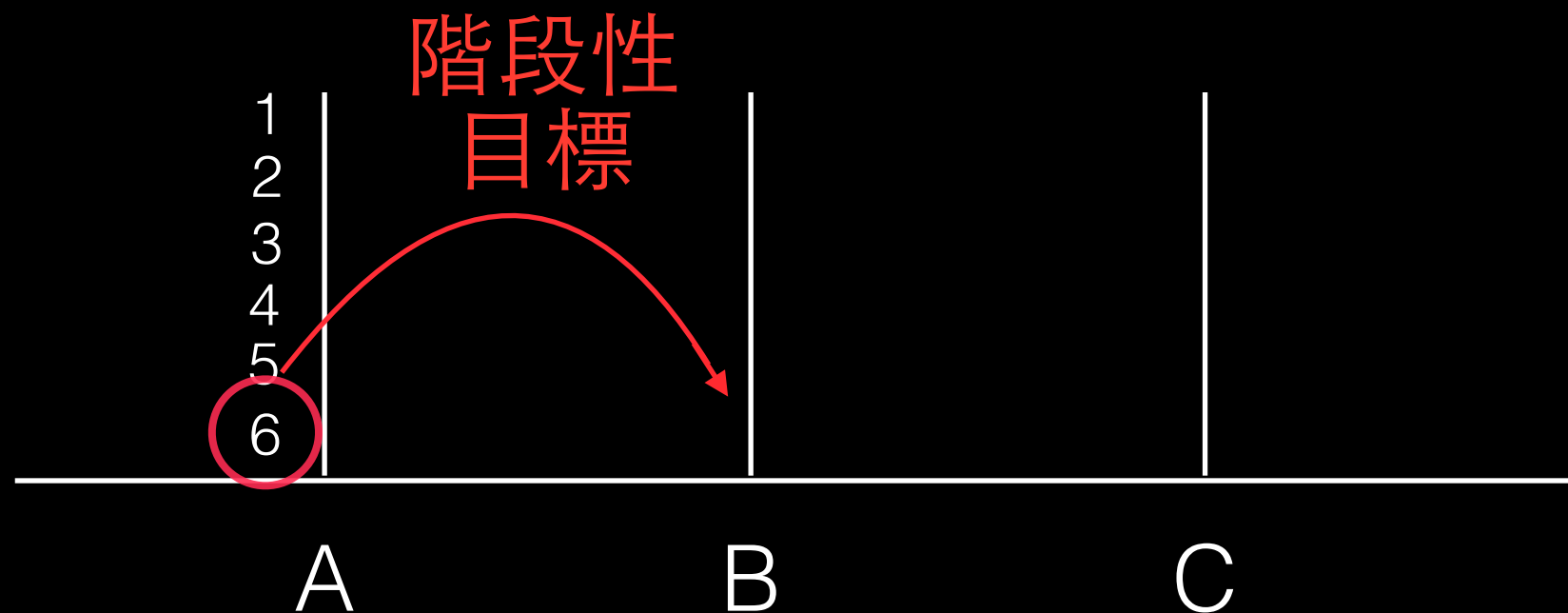
然後才能到達C柱子

目標：把最大的碟子移到C柱

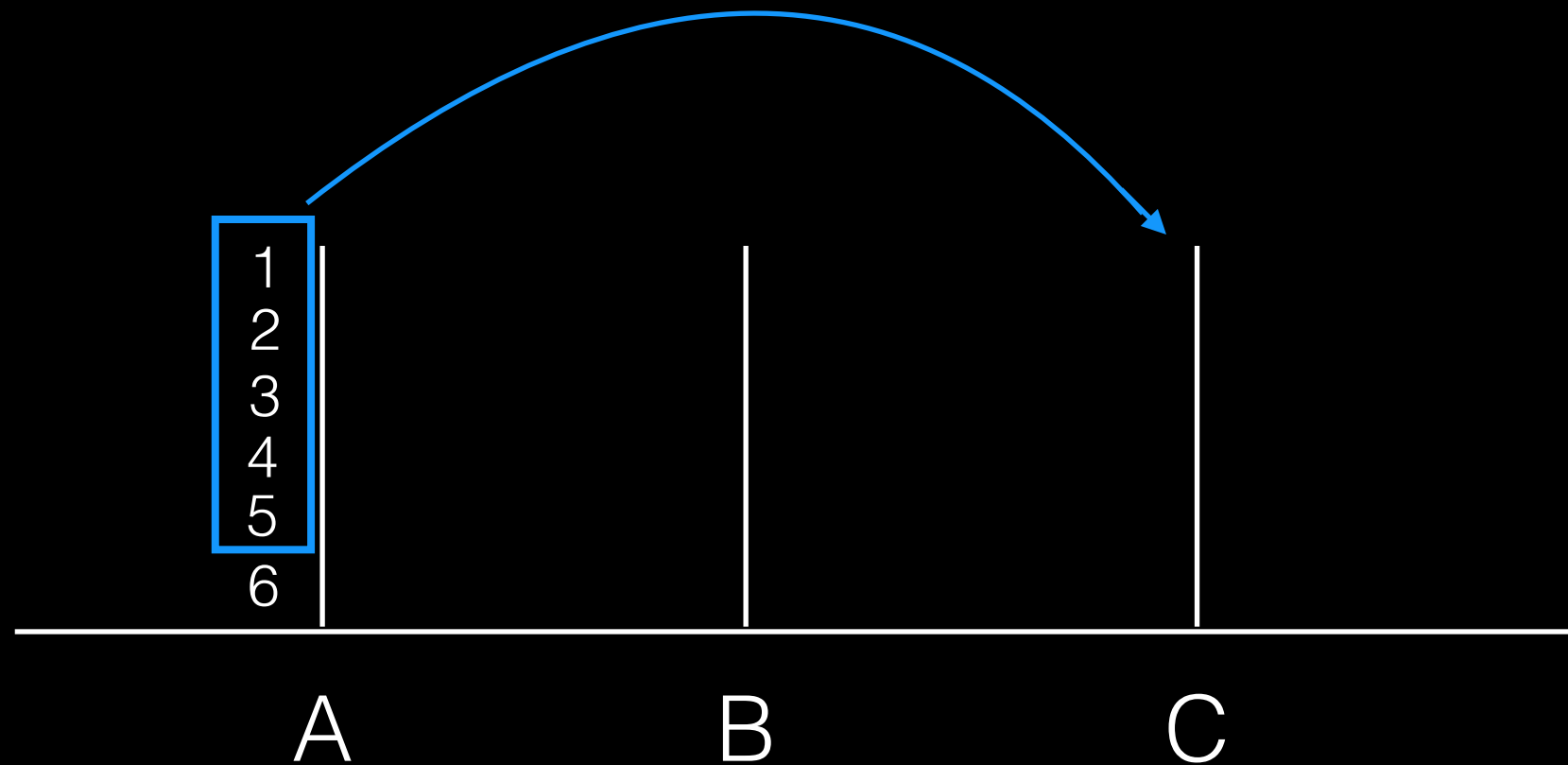


依照相同的概念

階段性目標：最大的碟子，要先放在柱子B

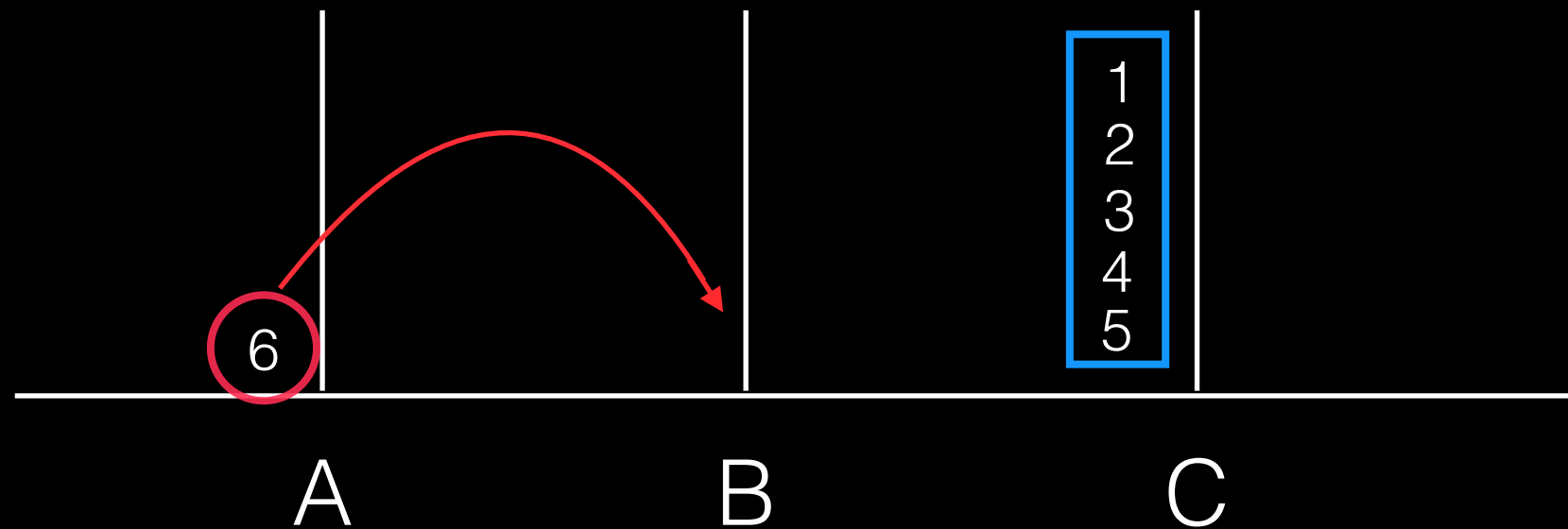


所以：上面的碟子，要先放在柱子C



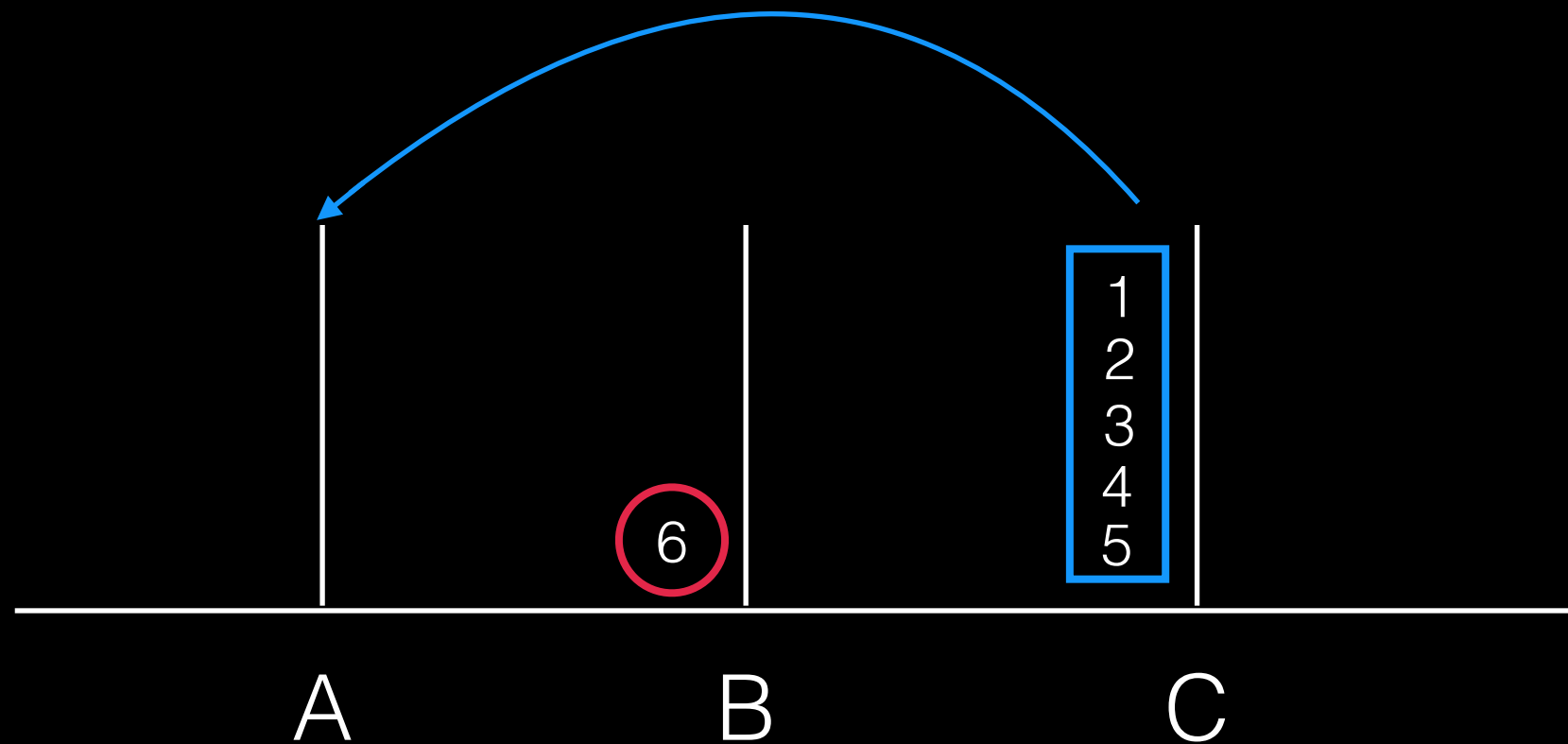
先把 $n-1$ 個碟子從A放到C
 $n = 6$

接著：把最大的碟子放在柱子B



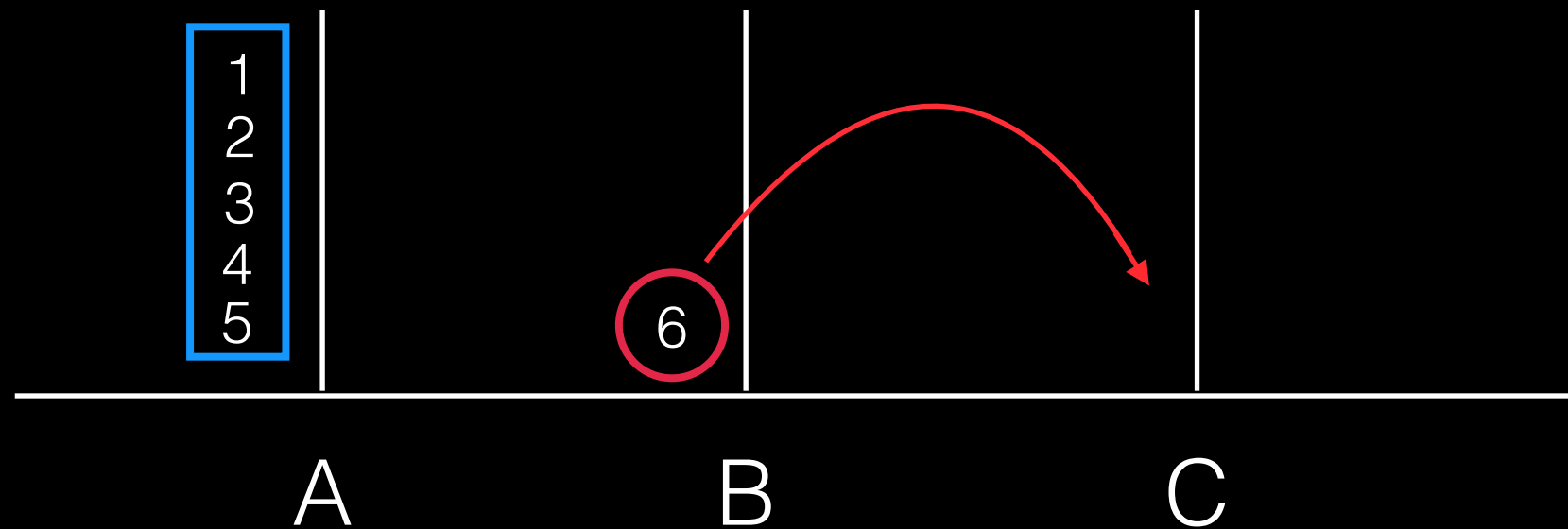
先把1個碟子從A放到B
 $n = 6$

接著：
把上面的碟子，從柱子C放到柱子B



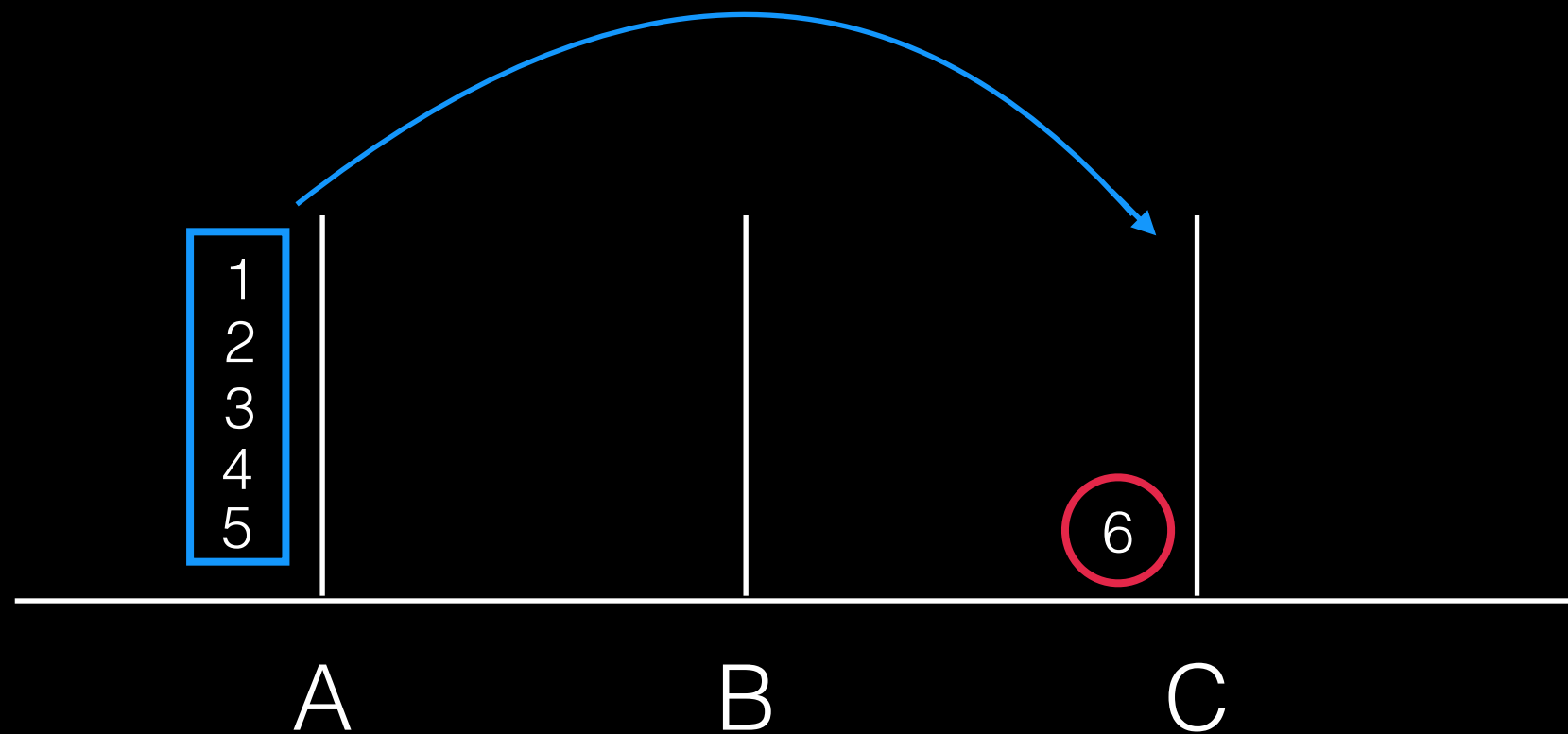
把 $n-1$ 個碟子從C放到A
 $n = 6$

接著：
把最大的碟子，從柱子B放到柱子C



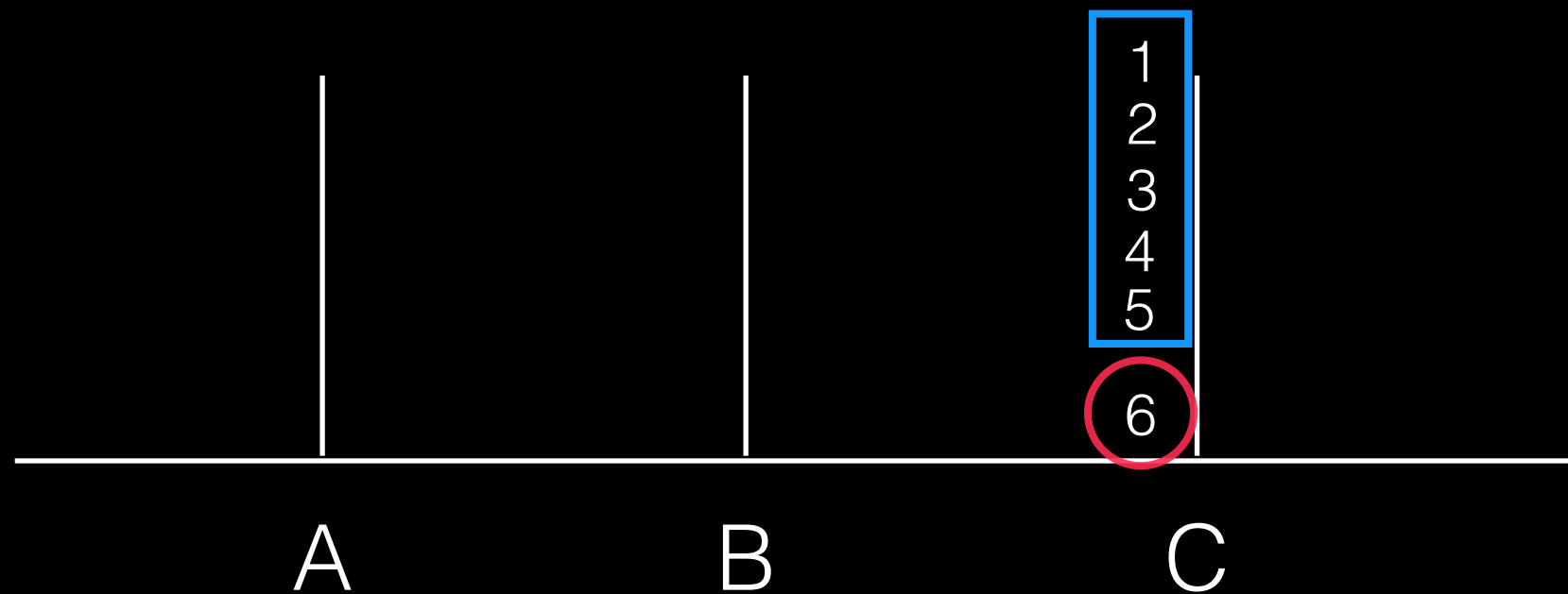
把1個碟子從B放到C
 $n = 6$

接著：
把上面的碟子，從柱子A放到柱子C



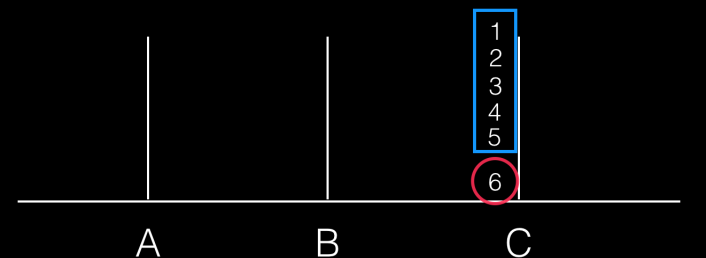
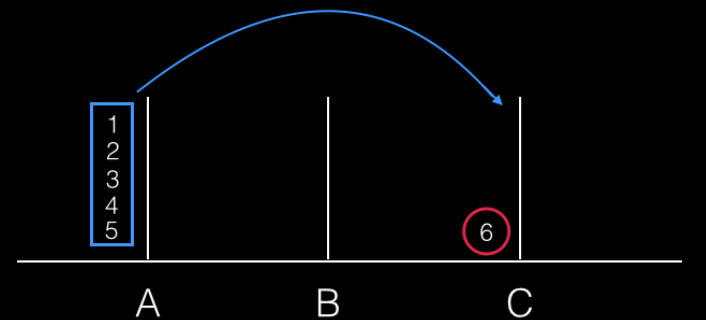
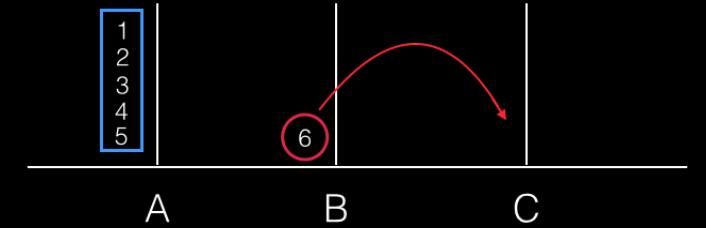
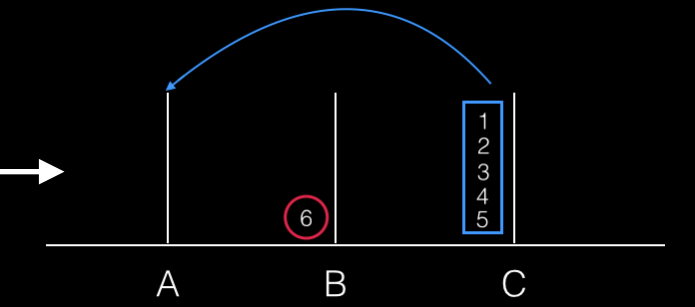
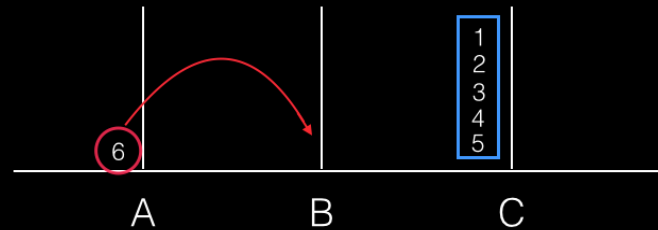
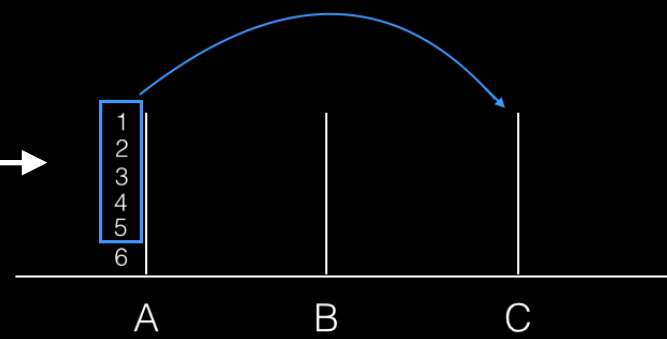
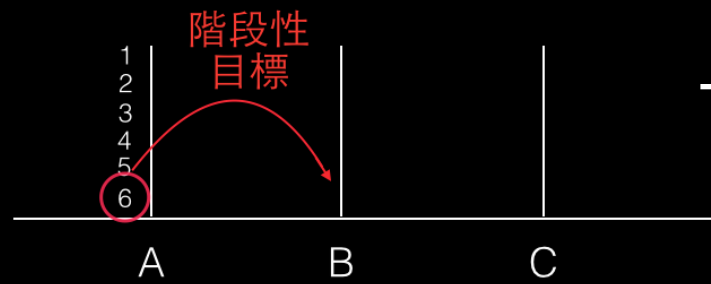
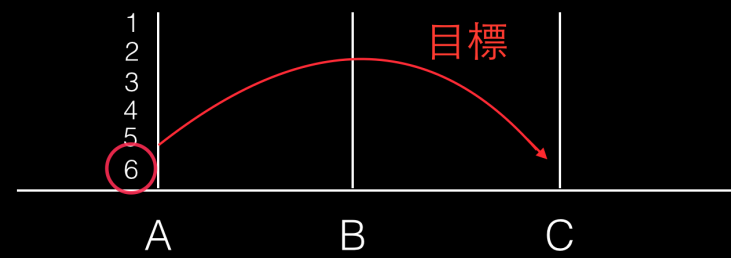
把 $n-1$ 個碟子從A放到C
 $n = 6$

完成



$n = 6$

整個流程



限制級(?)

河內塔的演算法

$$T(n) = 3T(n-1) + 2$$

n-1個
碟子

移動三次

1個碟子
移動兩次

就會變得
非常直覺

程式碼

- demo

參考資料

- 發現演算法
- Towers of Hanoi - Rosetta Code
- Infinite Loop: 【演算】 河內塔 - Tower of Hanoi