

recursive

5/15, hydai

「To iterate is human, to recurse, divine.」

-L. Peter Deutsch

復習

- 驟迴函式就是會自己呼叫自己的函式
- 驅迴必要的東西：
 - 條件關係：衍生呼叫的關係
 - 中止條件：決定何時結束
 - 加速：把算過的東西存起來

TOJ 33：費式數列

- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$, $n \geq 2$

TOJ 33：費式數列

- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$, $n \geq 2$
- $F_2 = F_1 + F_0 = 2$

TOJ 33：費式數列

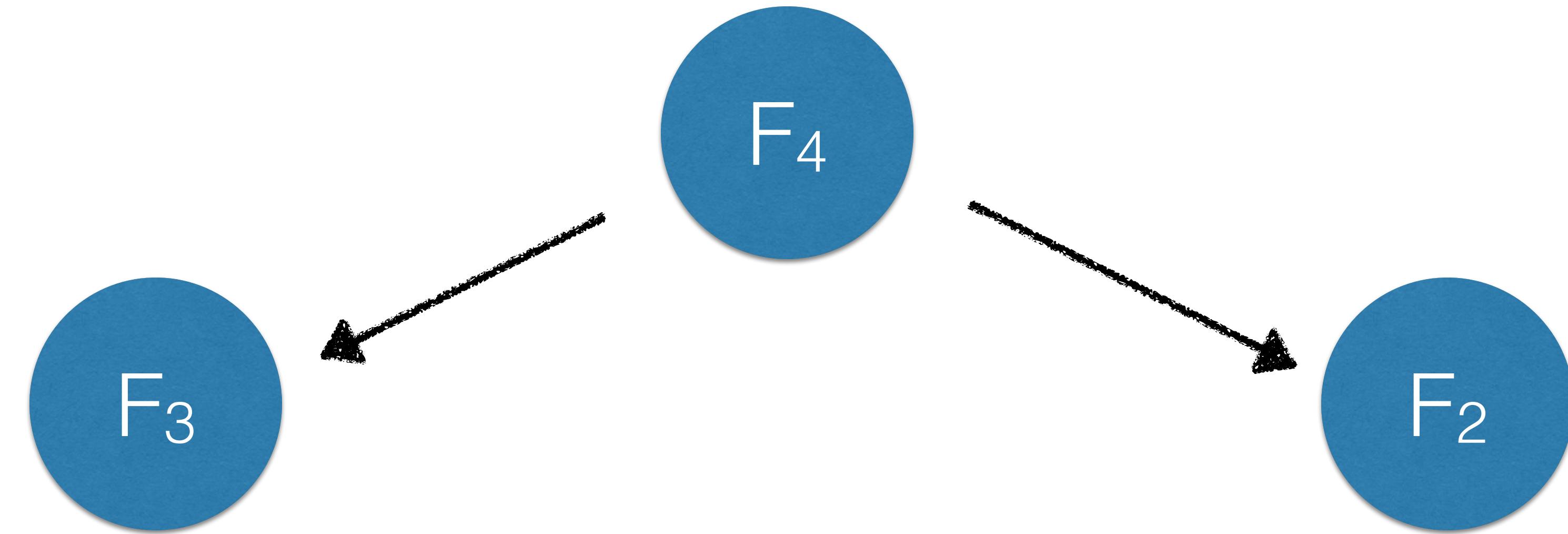
- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$, $n \geq 2$
- $F_2 = F_1 + F_0 = 2$
- $F_3 = F_2 + F_1 = 3$

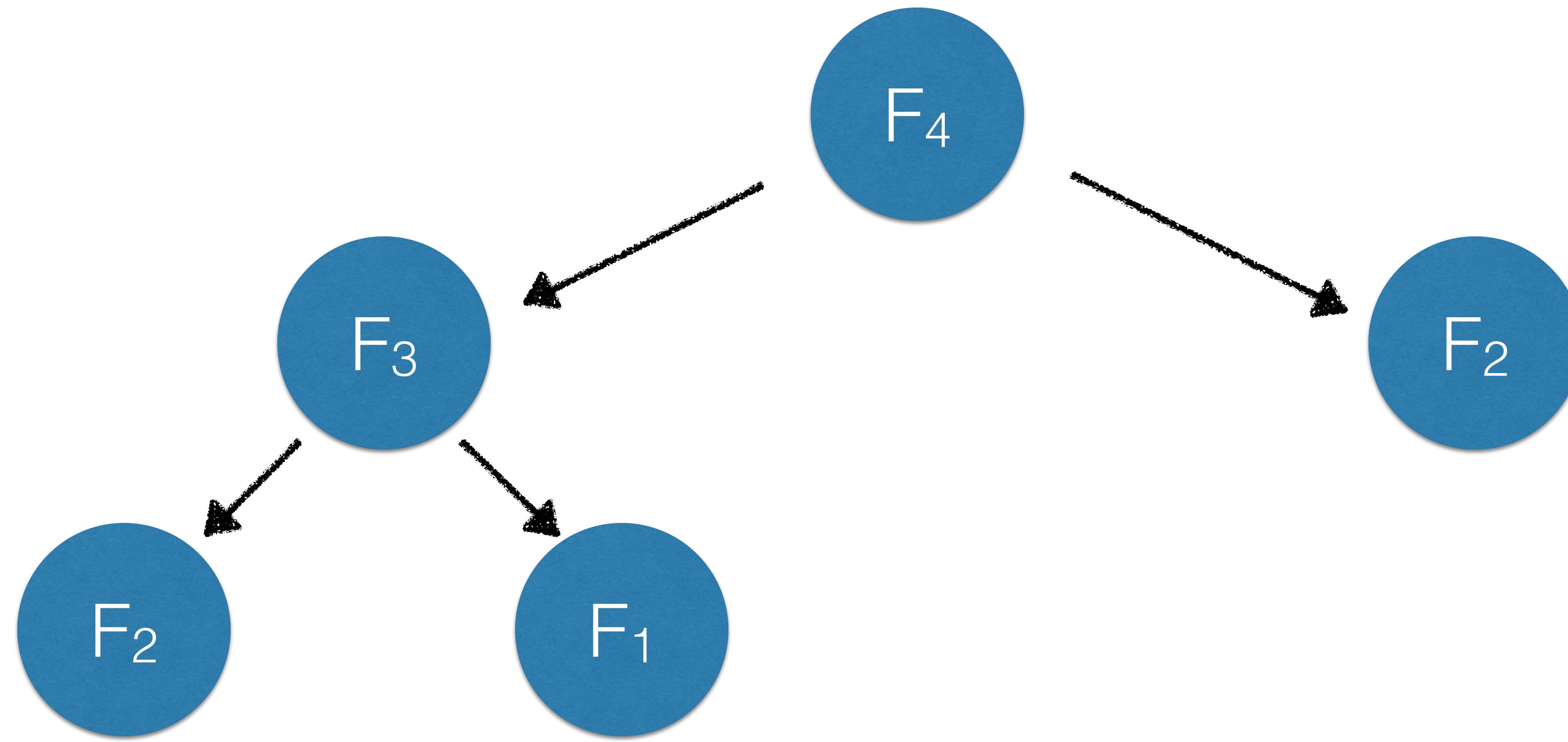


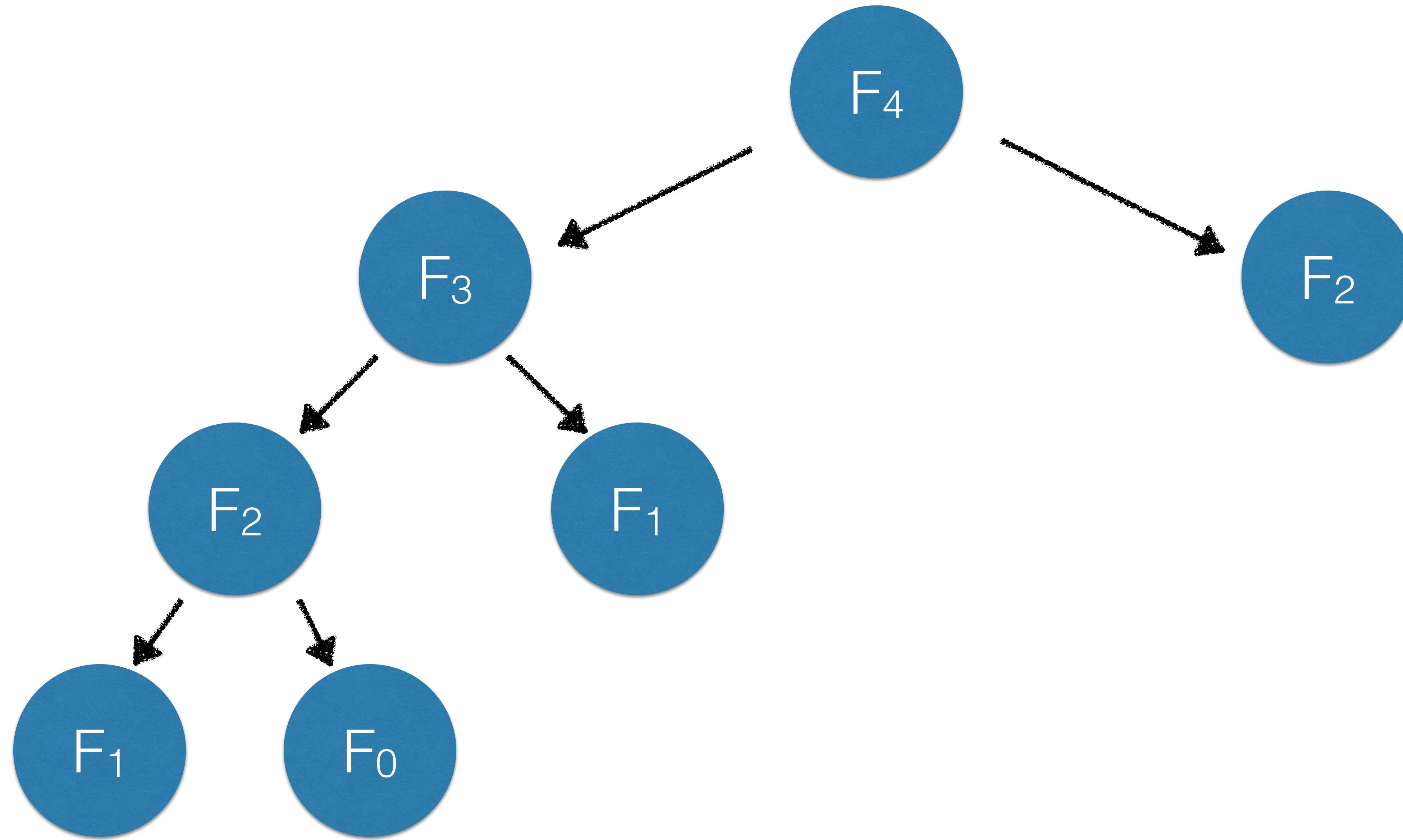
TOJ 33：費式數列

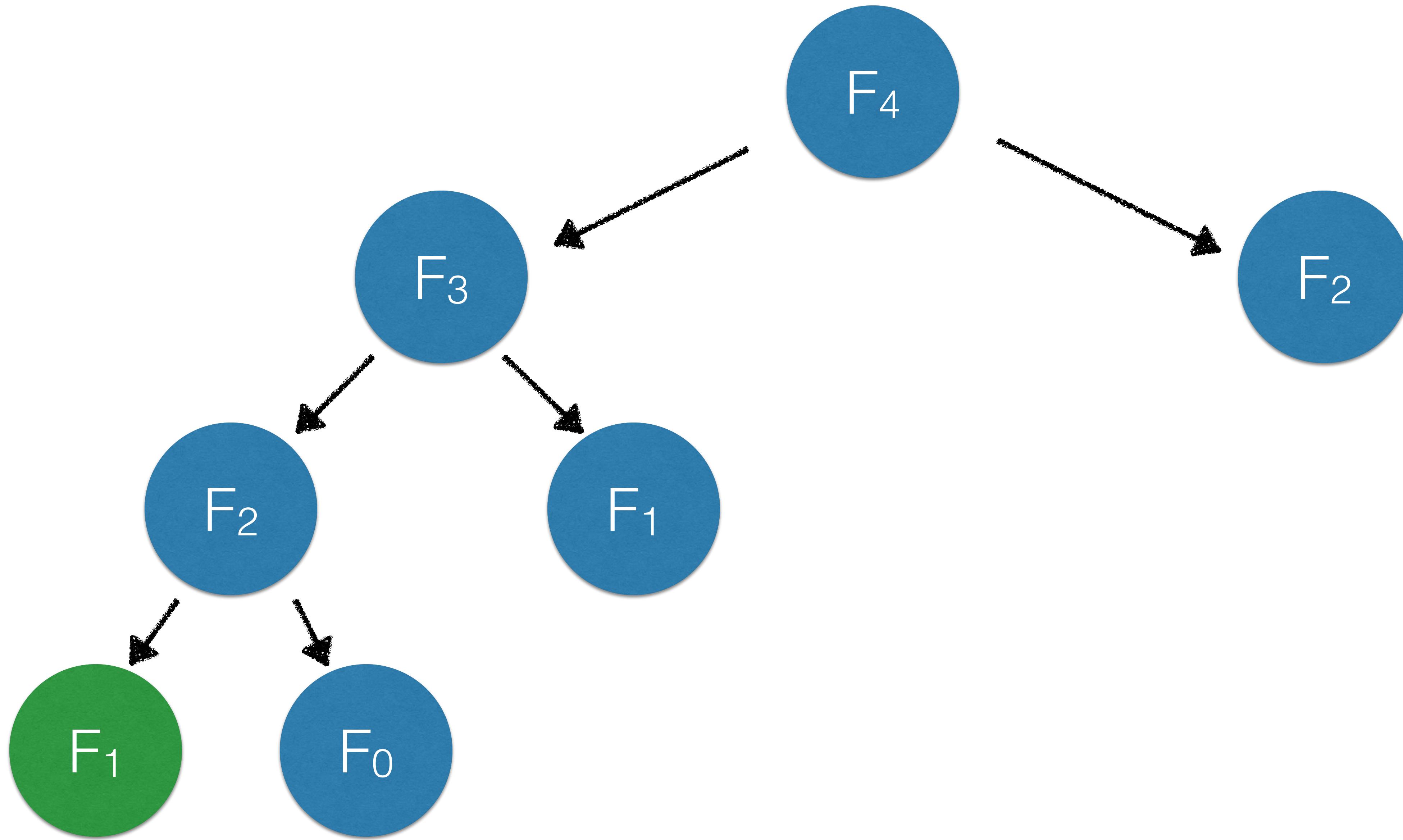
- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$, $n \geq 2$
- $F_2 = F_1 + F_0 = 2$
- $F_3 = F_2 + F_1 = 3$
- $F_4 = F_3 + F_2 = 5$
- ...

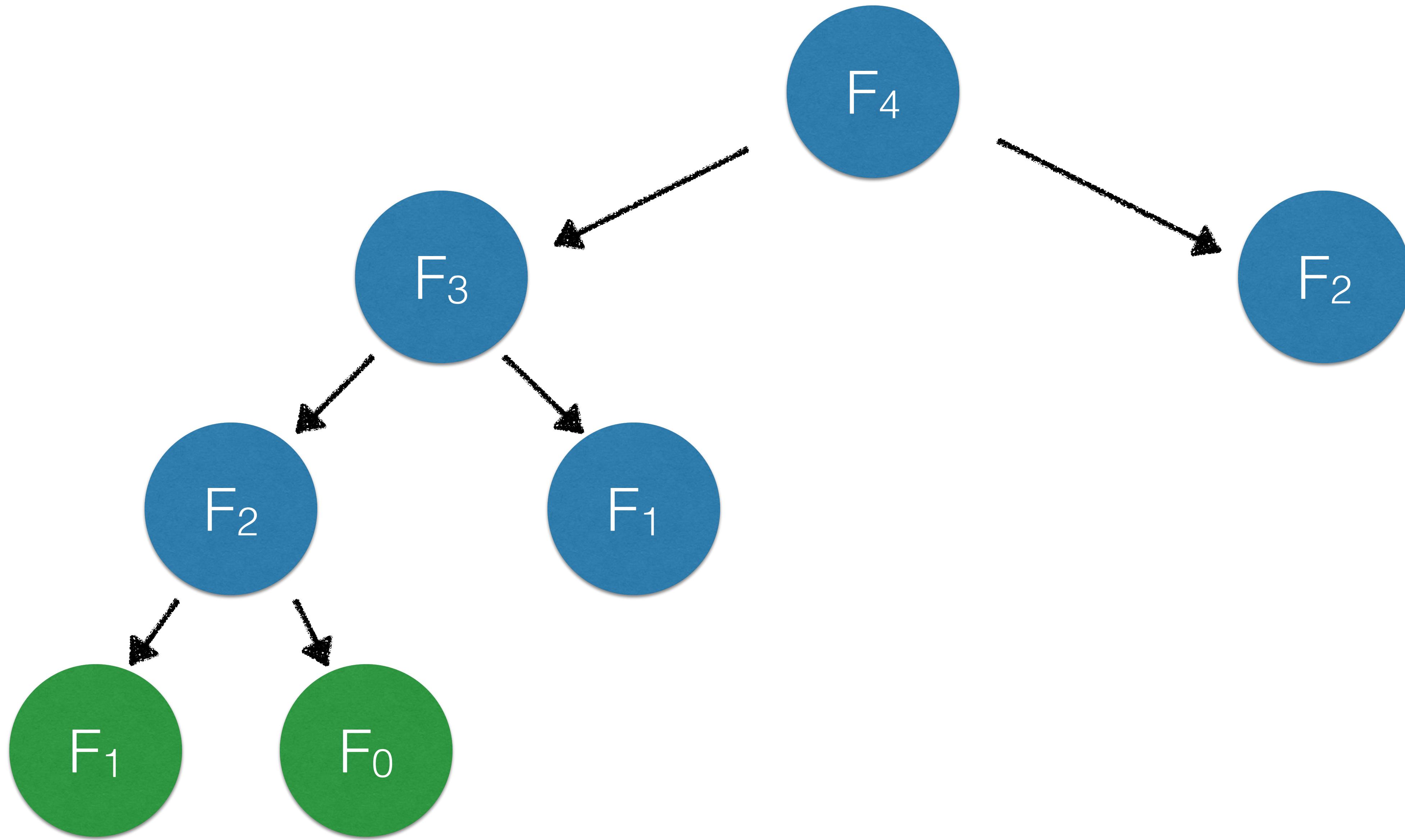
F_4

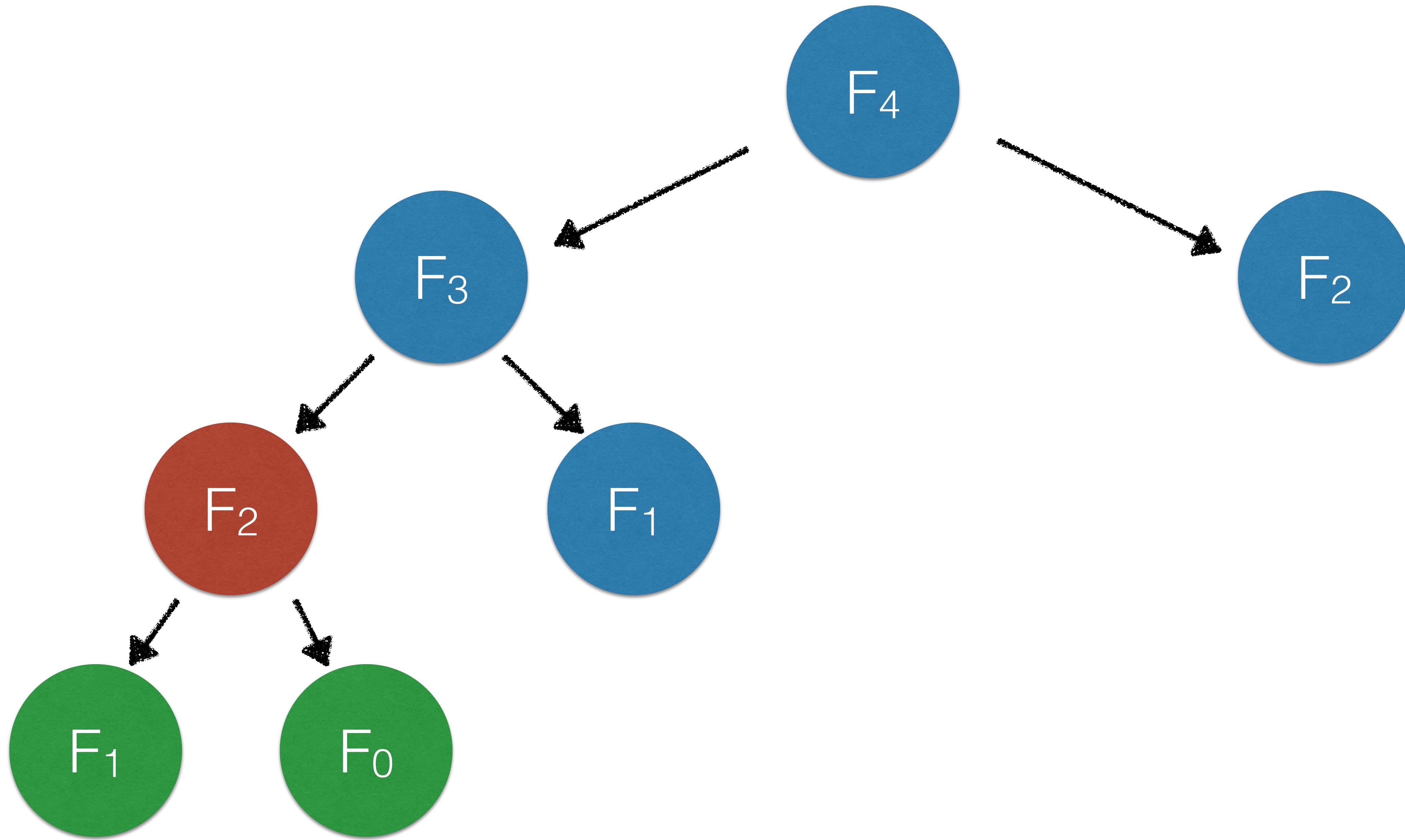


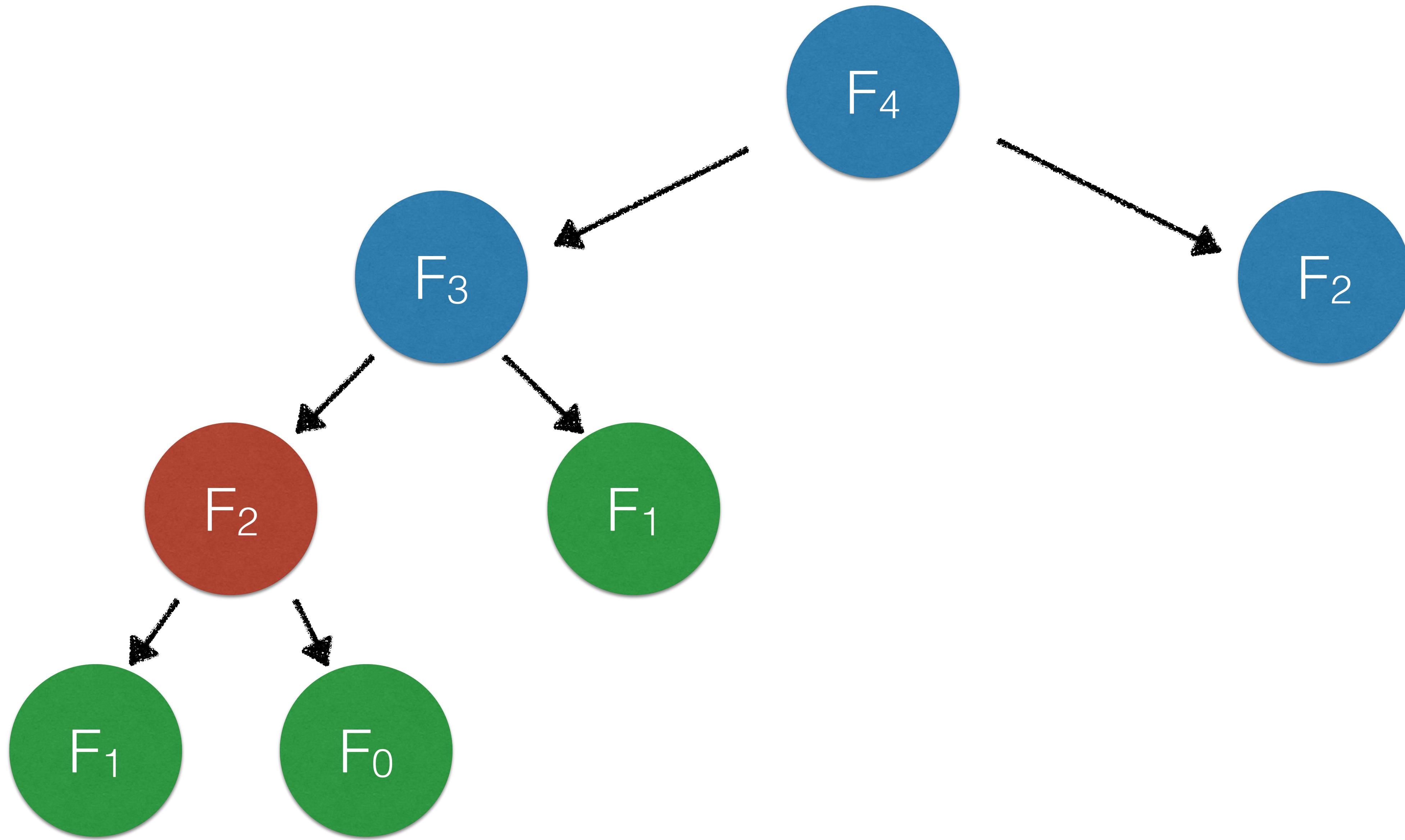


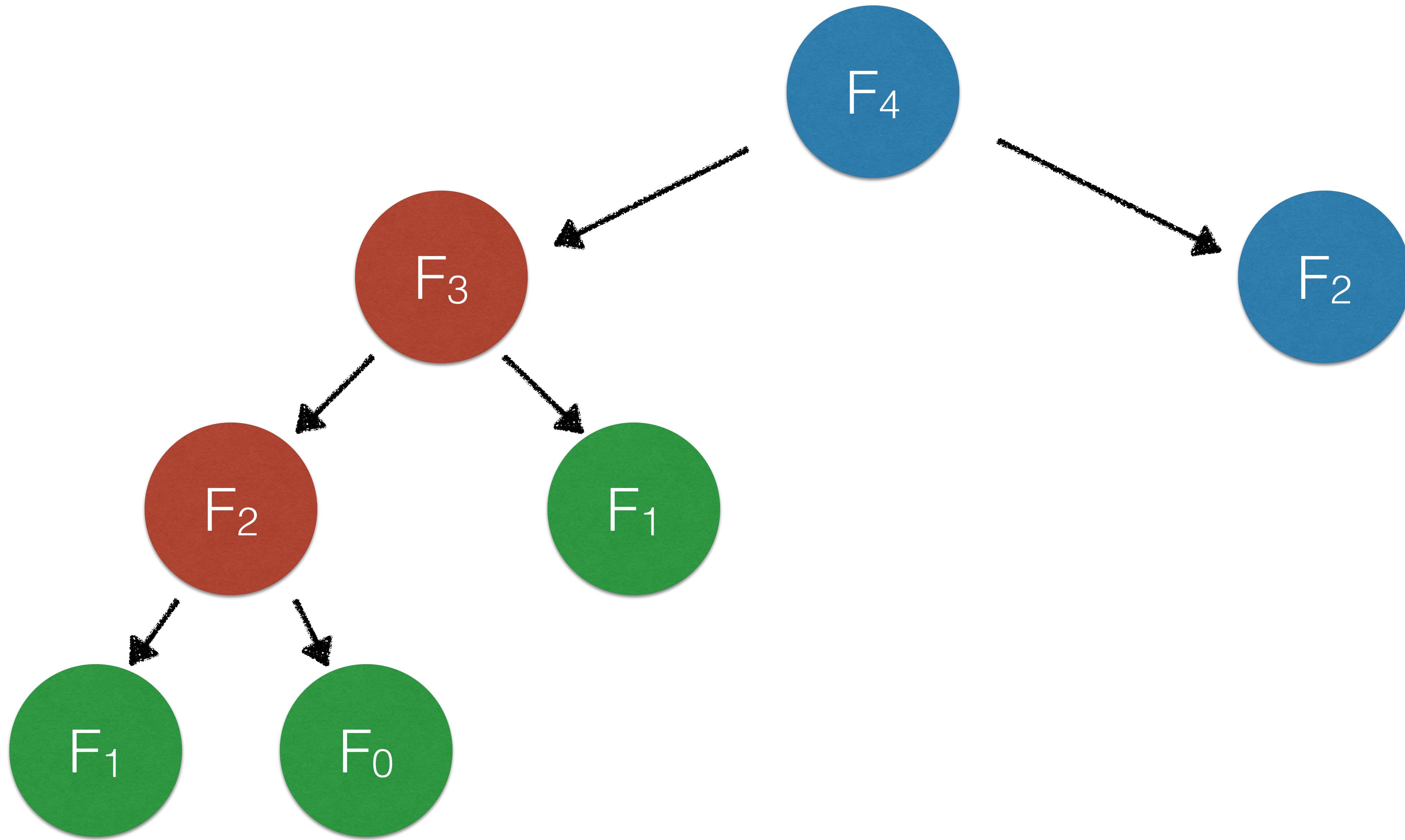


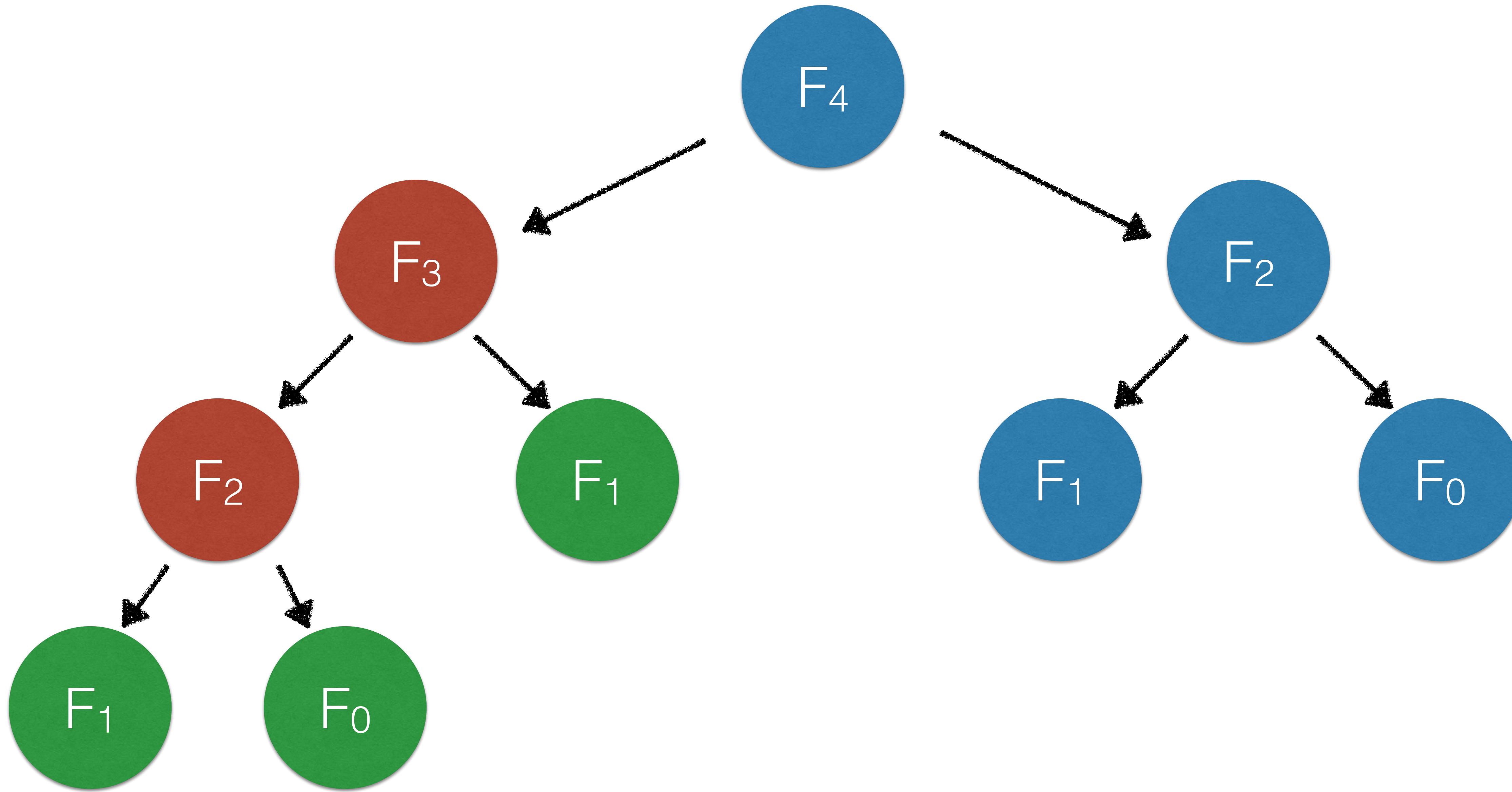


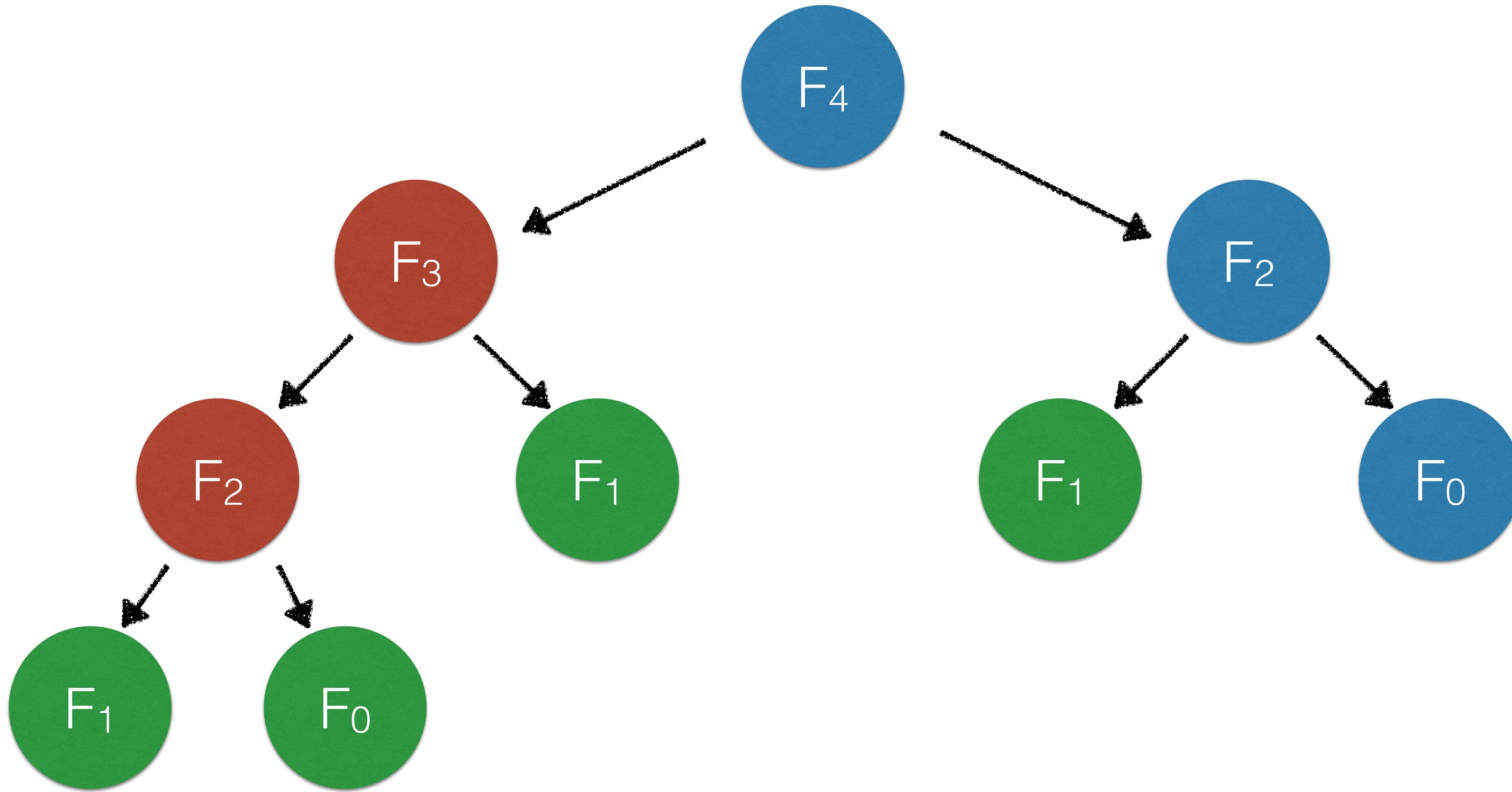


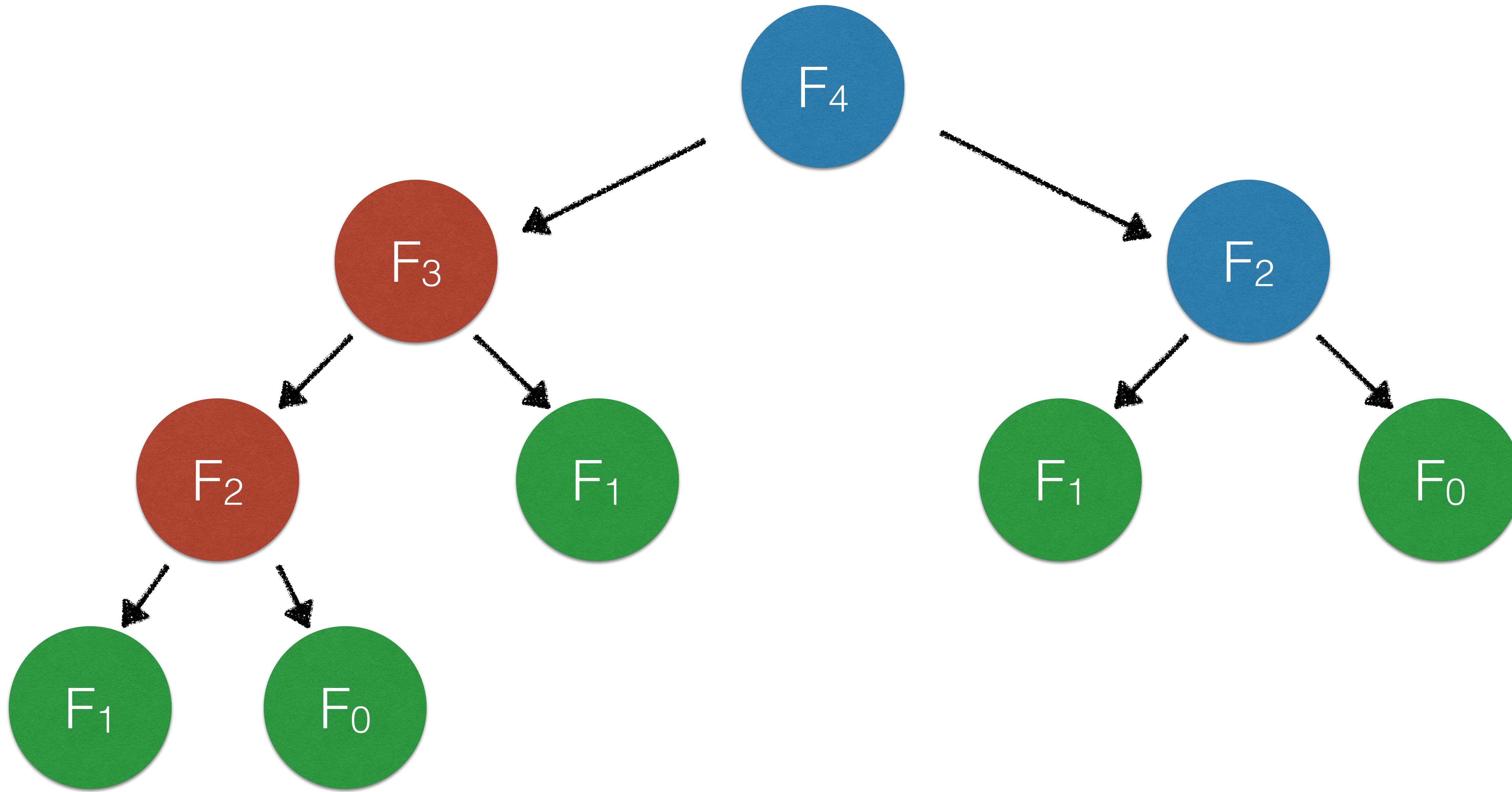


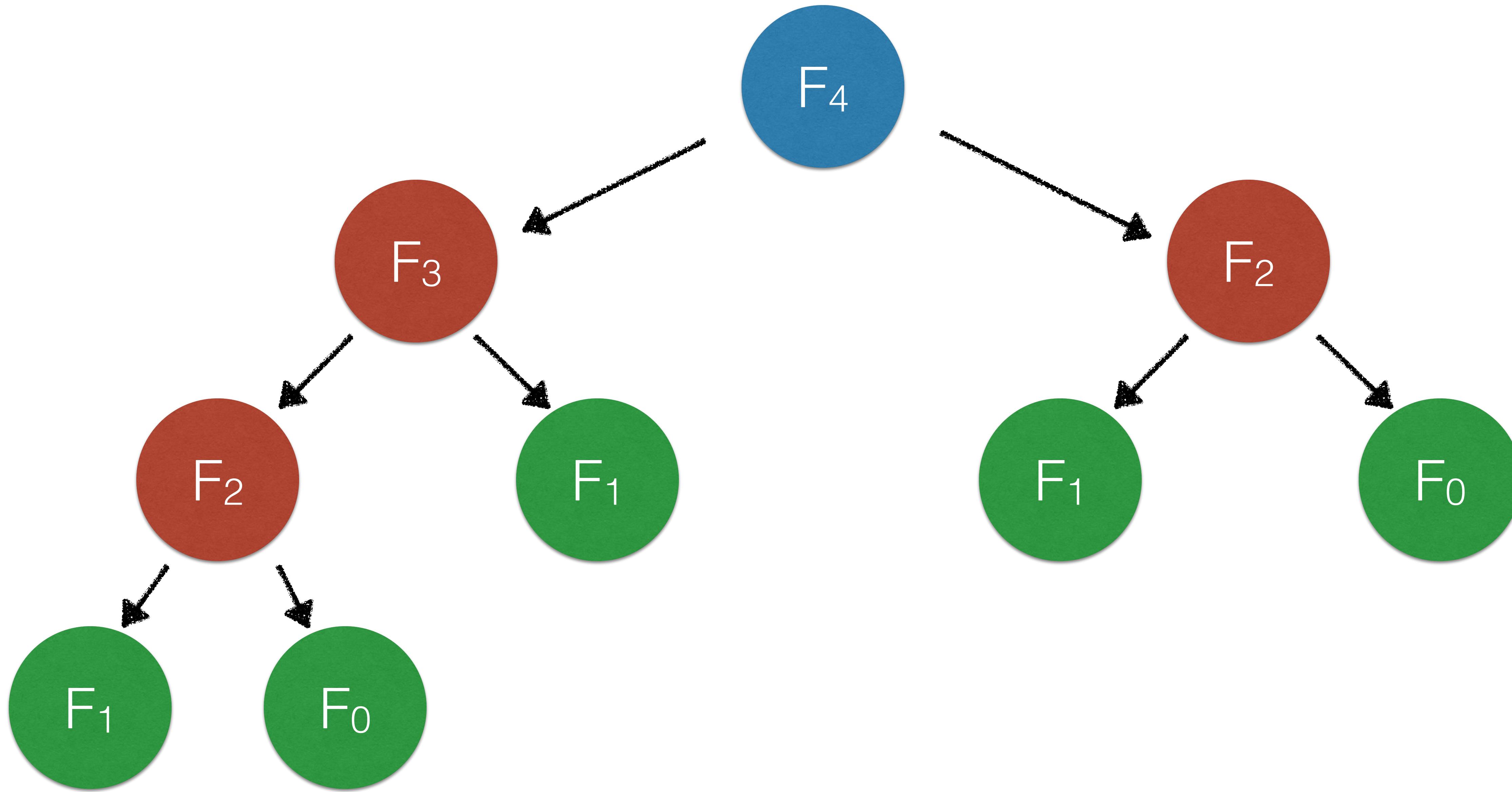


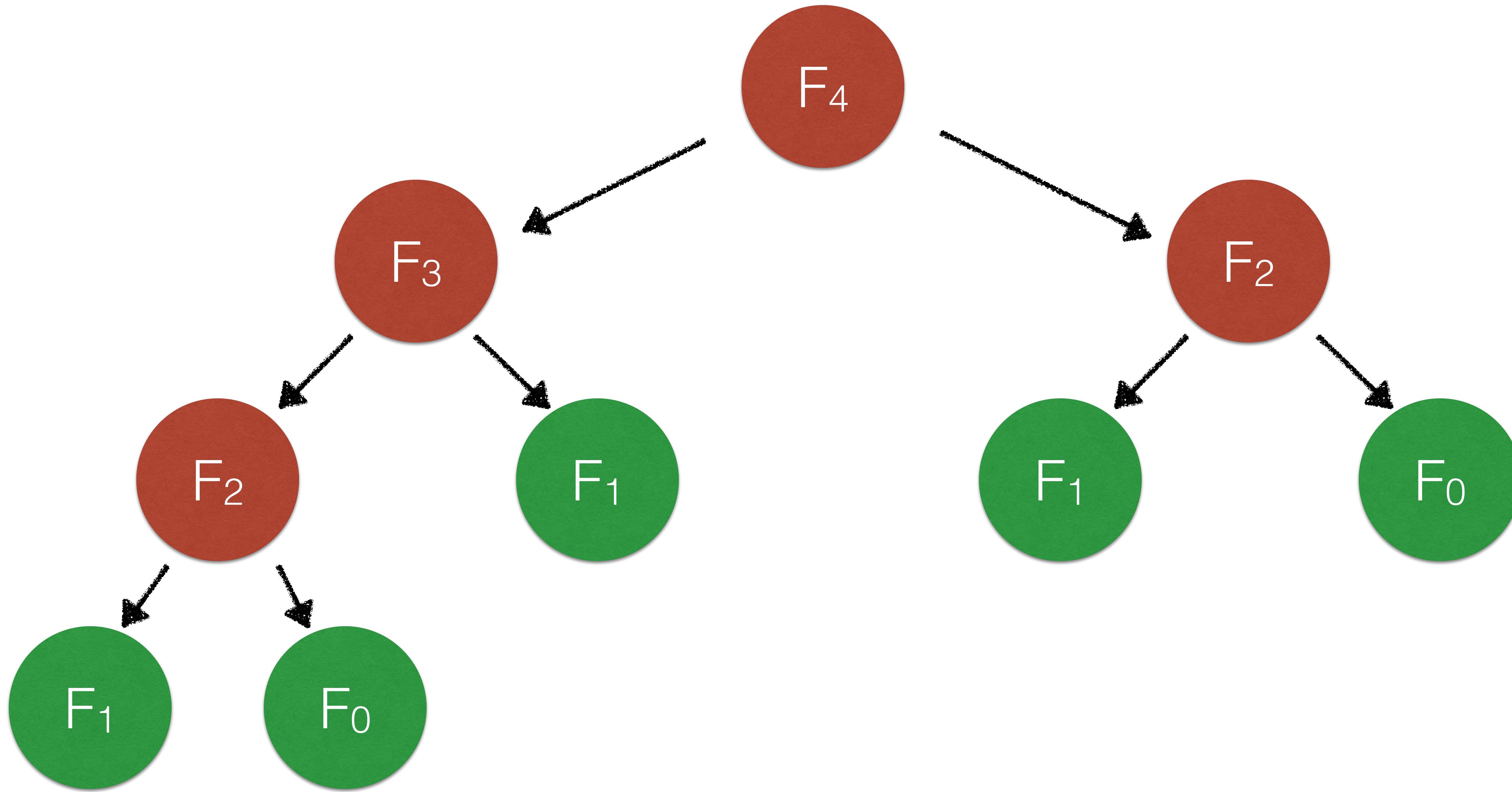












本週作業

- 153：文字轉轉轉
- 154：河內之塔

作業講解，文字轉轉轉

- 紿定一個排序好的字串（為大寫字母A-Z，且每個字元相異）
- 請你輸出這個字串的所有排列的可能（以字典序印出）
- Input:
- AB
- Output:
- AB
- BA

暫存的輸出字串

選取的文字

沒被選中的文字

AB

暫存的輸出字串

A

選取的文字

A

沒被選中的文字

AB

B

暫存的輸出字串

A

AB

選取的文字

A

AB

沒被選中的文字

AB

B

暫存的輸出字串

A

AB

A

選取的文字

A

AB

A

沒被選中的文字

AB

B

B

暫存的輸出字串

A

AB

A

選取的文字

A

AB

A

沒被選中的文字

AB

B

B

AB

暫存的輸出字串

A

AB

A

B

選取的文字

A

AB

A

B

沒被選中的文字

AB

B

B

AB

A

暫存的輸出字串

A

AB

A

B

BA

選取的文字

A

AB

A

B

AB

沒被選中的文字

AB

B

B

AB

A

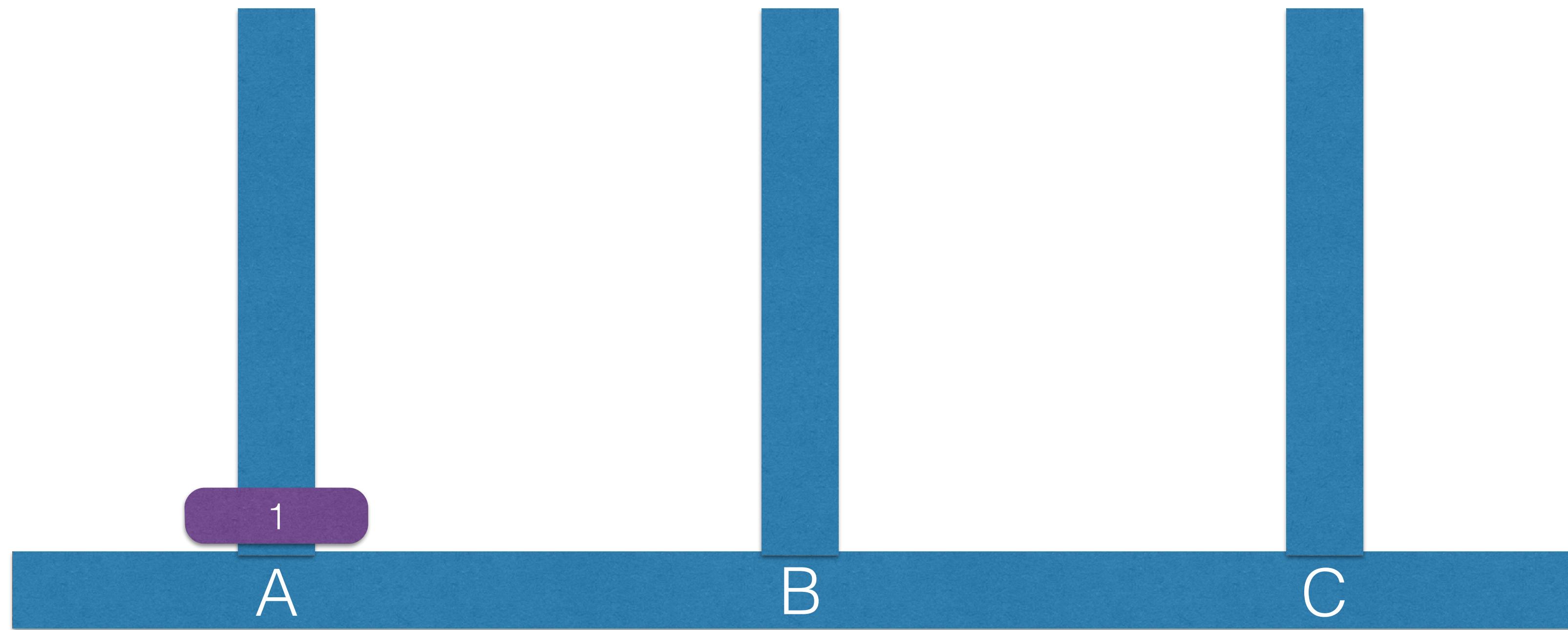
該怎麼想呢？

- 每一層都需要決定要選哪一個字母
 - 避開已經選到的字母
 - 測試每個可以選的字母，接著進入下一層
- 如果全部都選到以後，就可以把答案印出來，並退回去上一層
- 重複以上步驟直到字母全被測試過

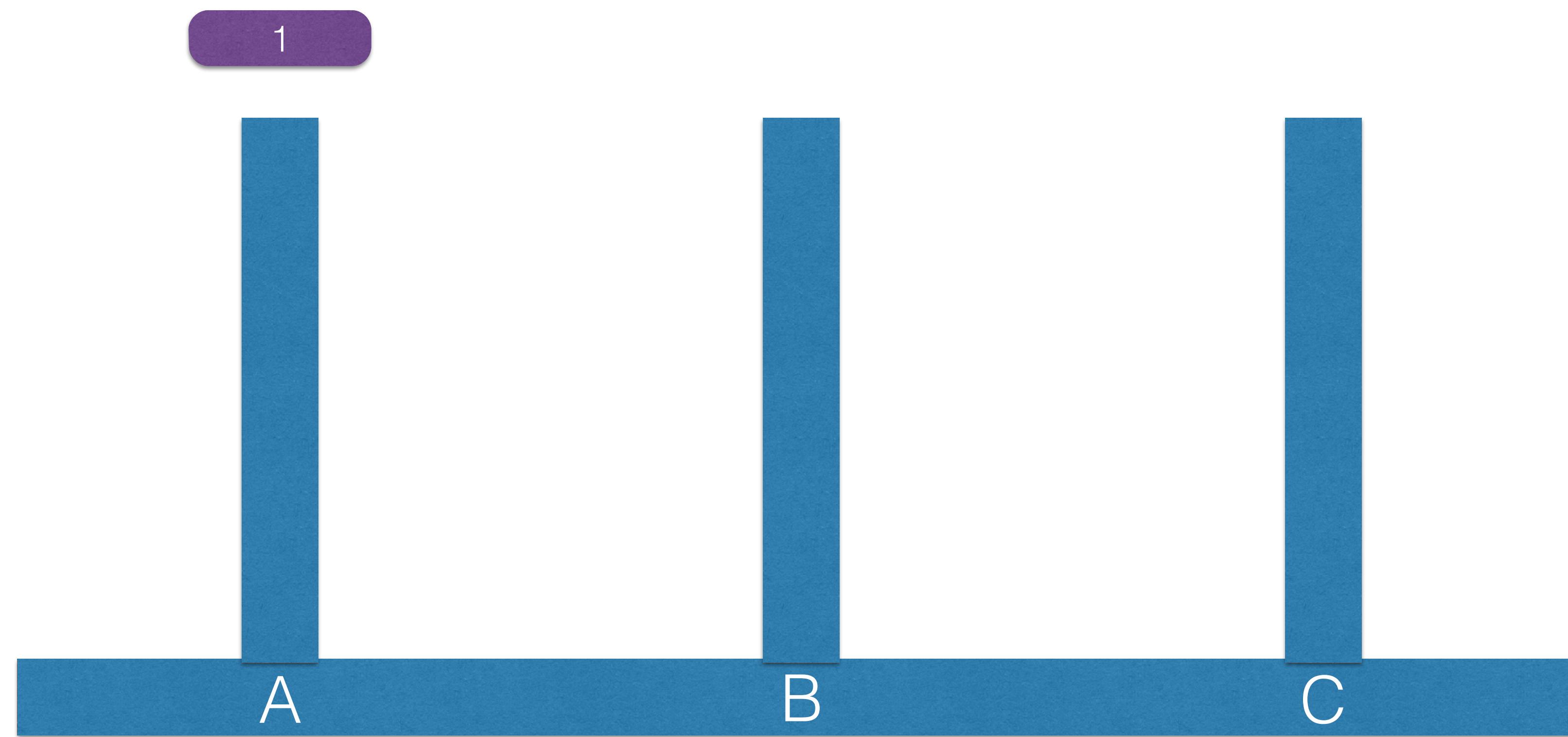
作業講解，河內塔

- 盤子有 N 個，每個大小都不一樣
- 只有小的能疊在大的上頭
- 起始，所有盤子都在最左邊
- 最終要把所有盤子放到最右邊

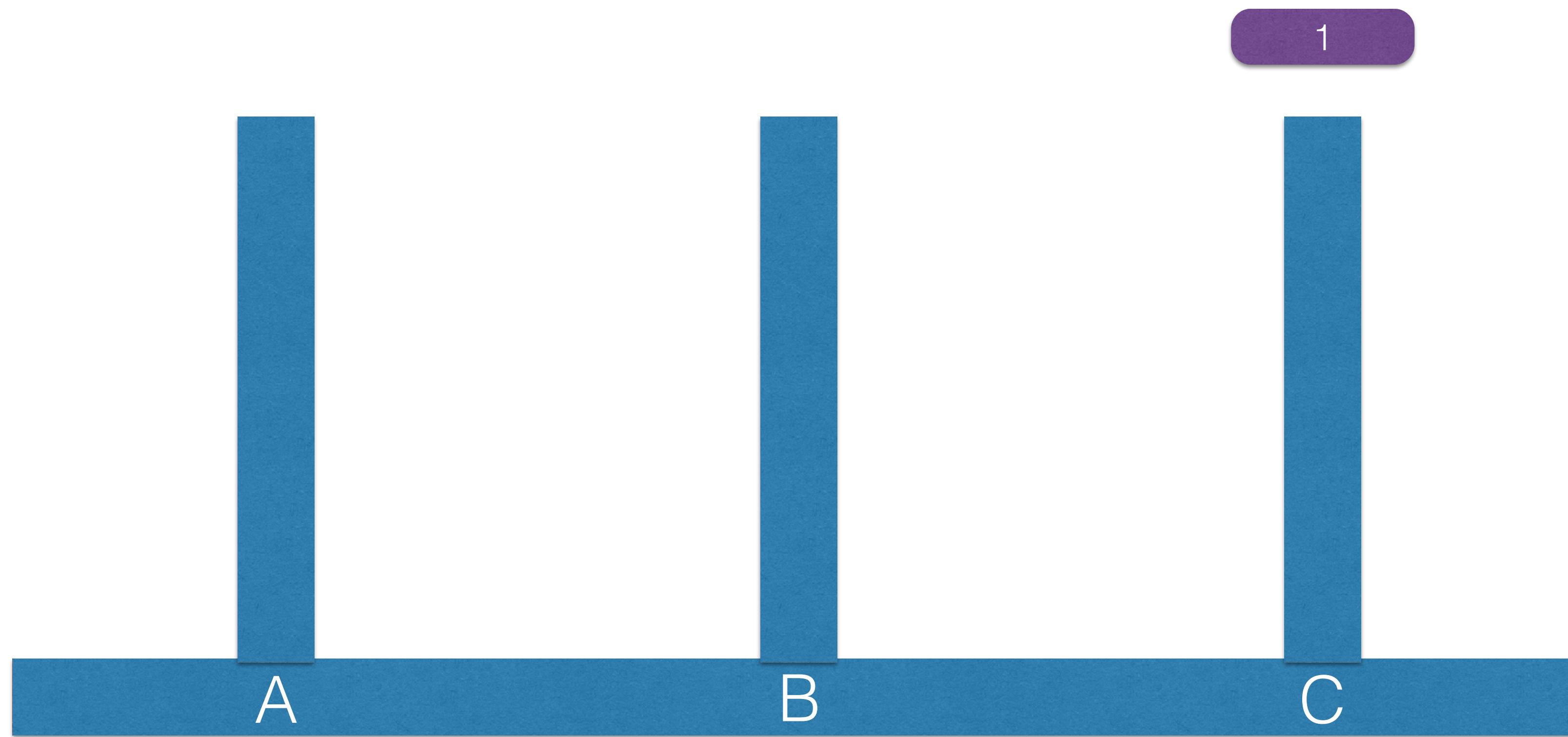
N = 1



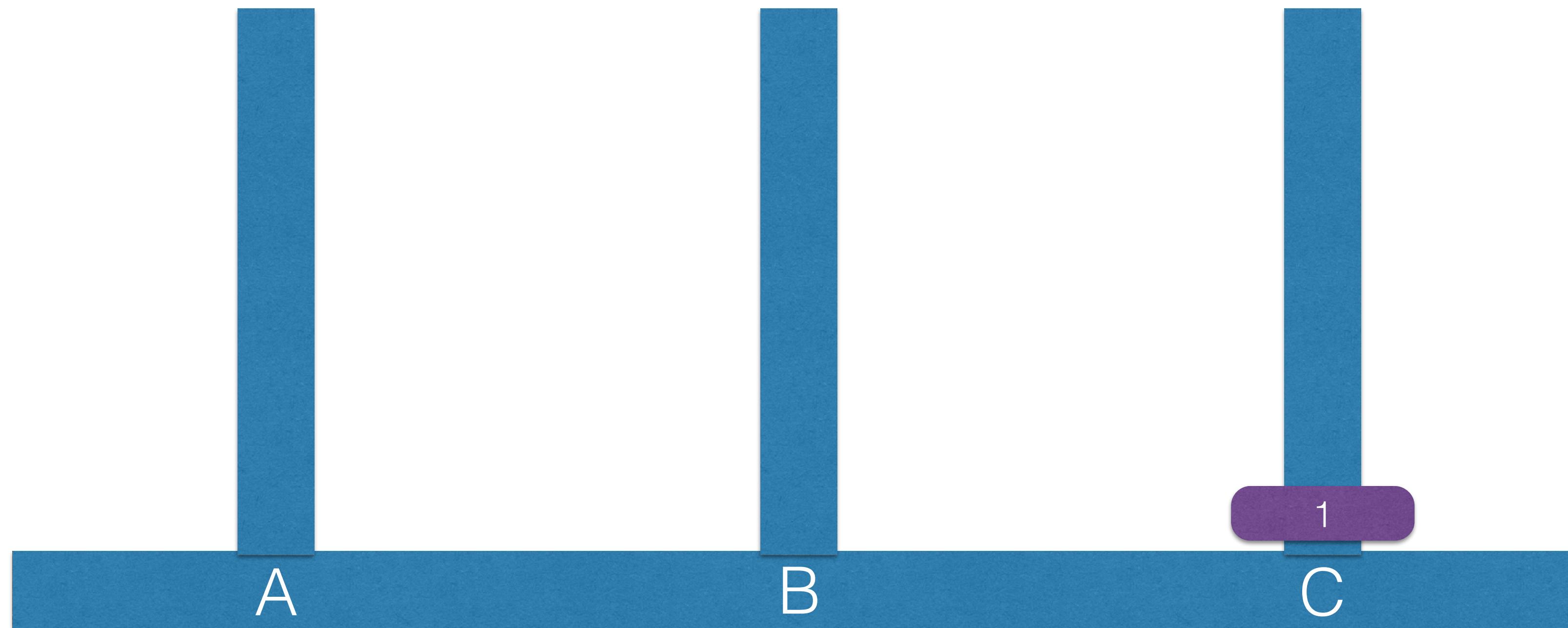
N = 1



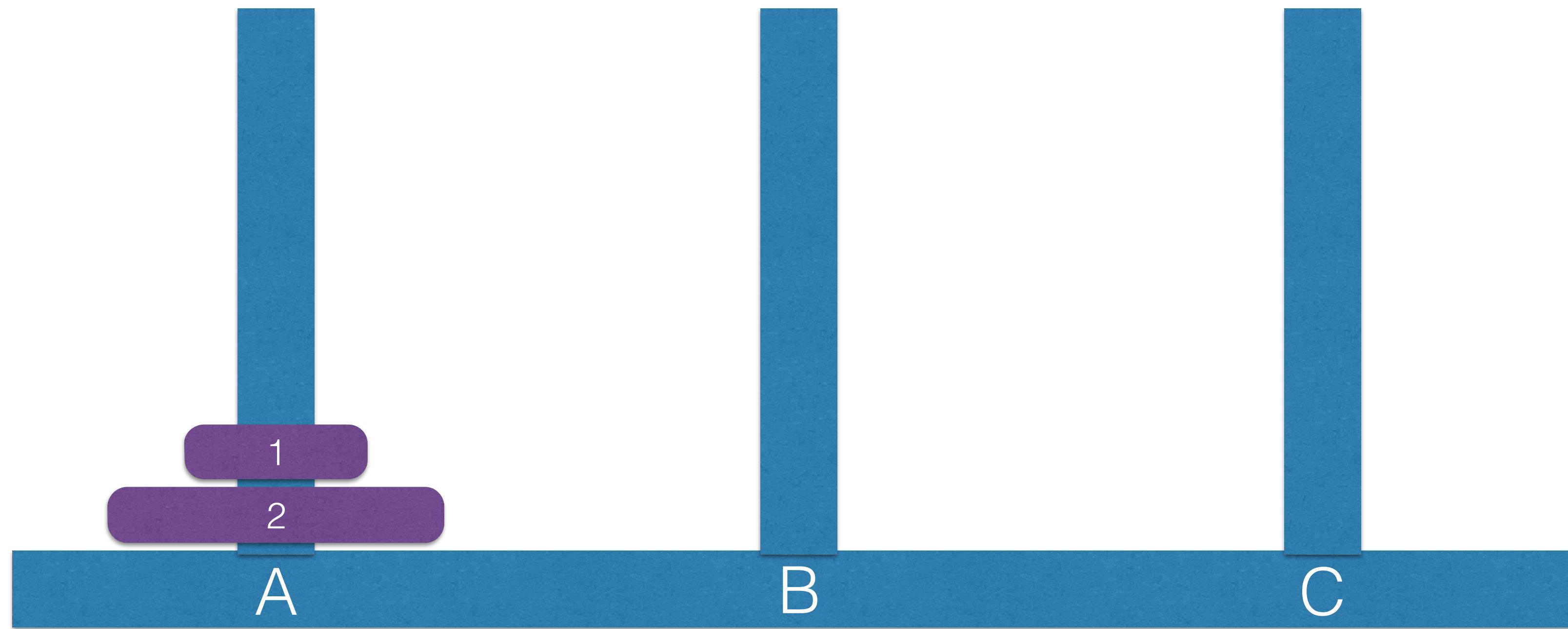
N = 1



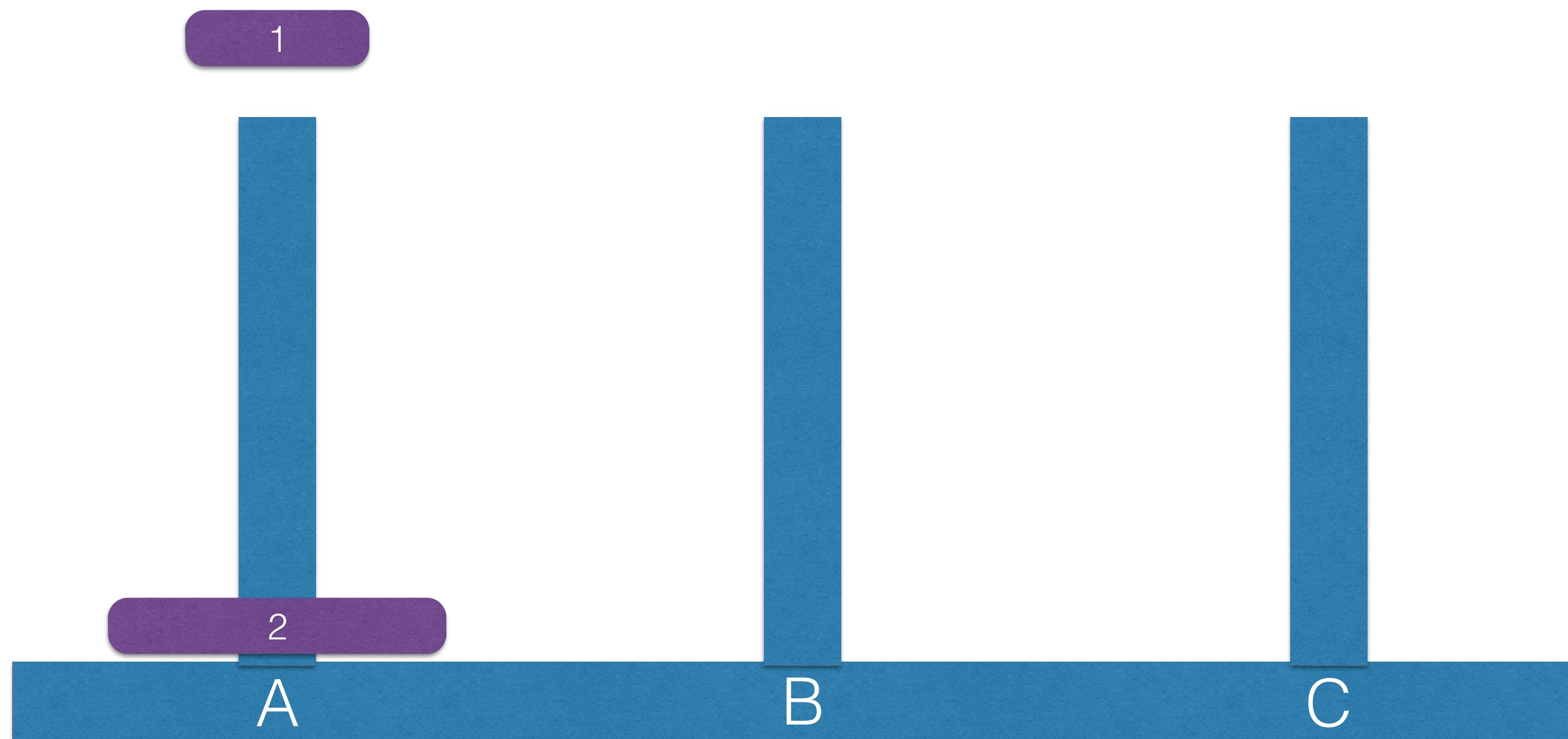
N = 1



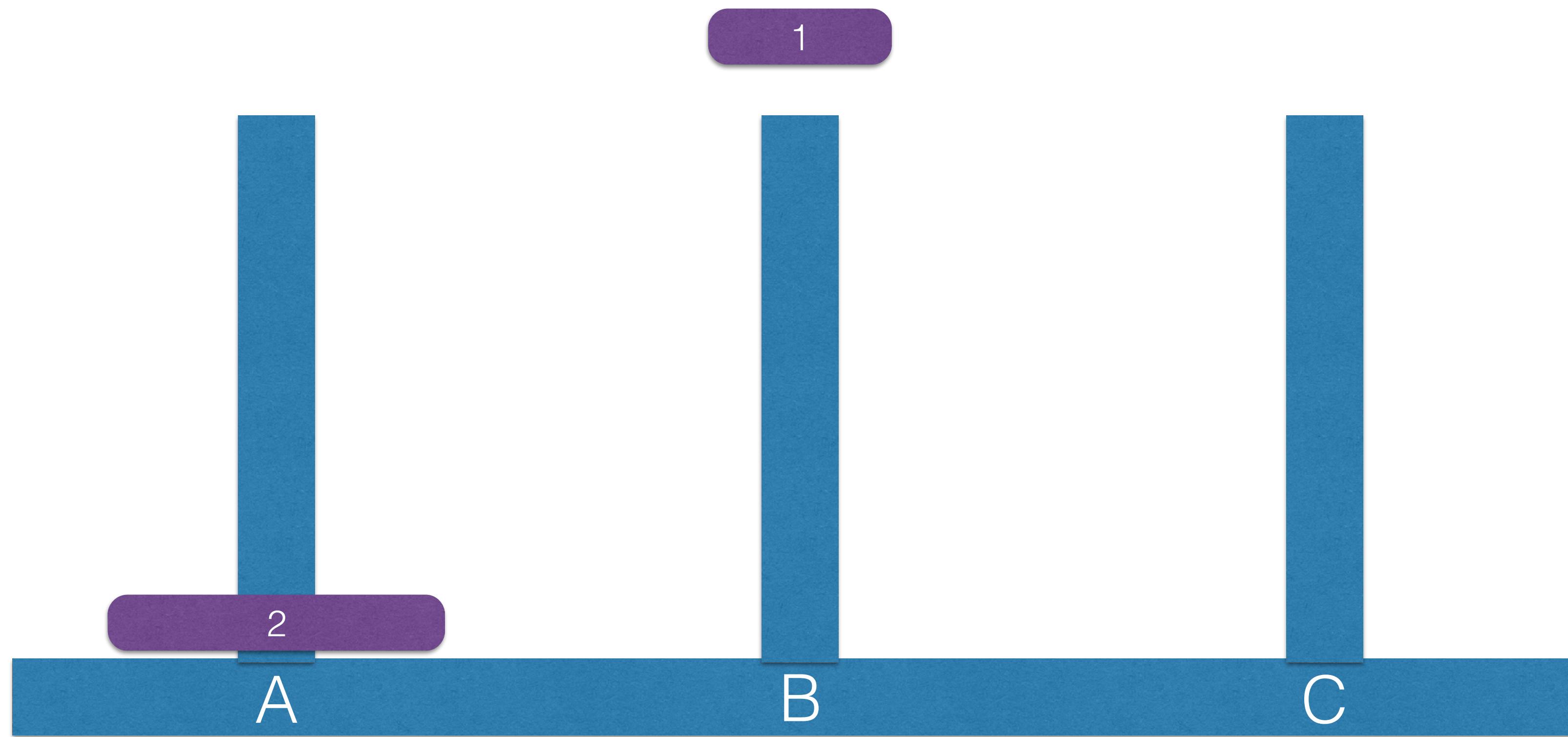
N = 2



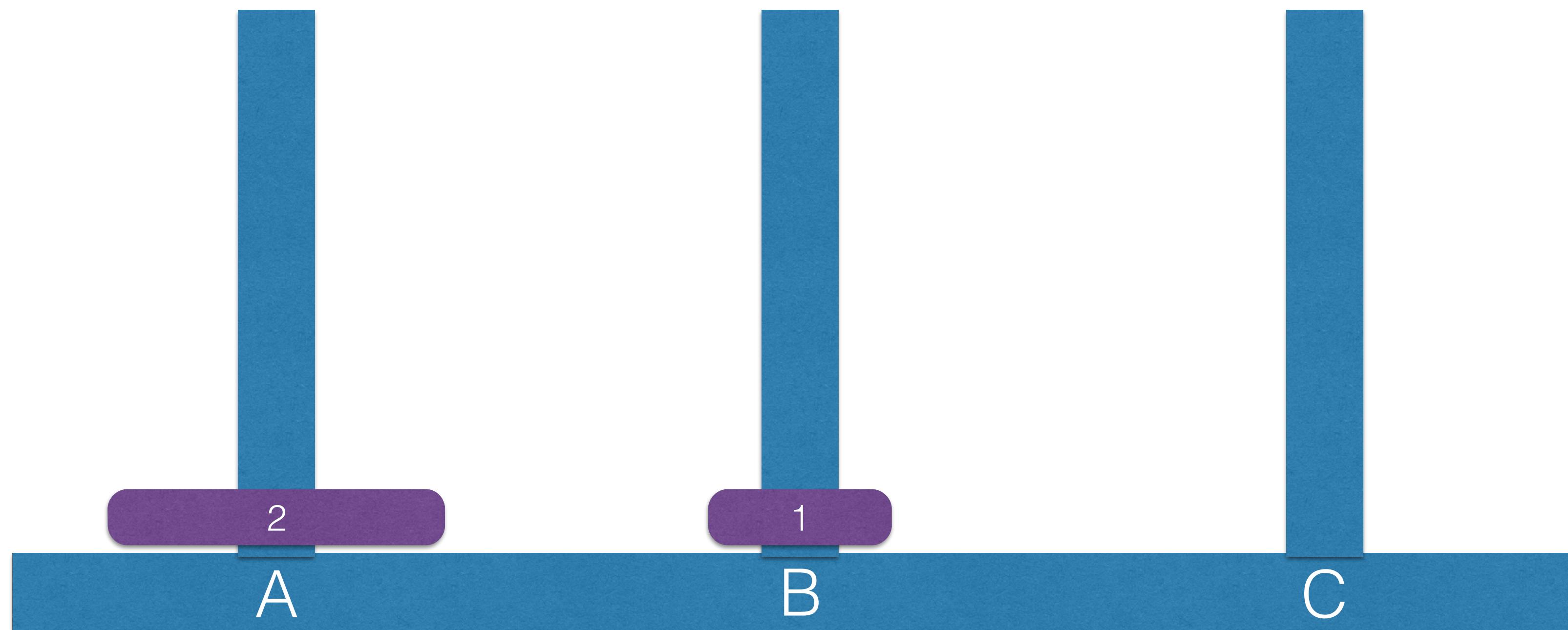
N = 2



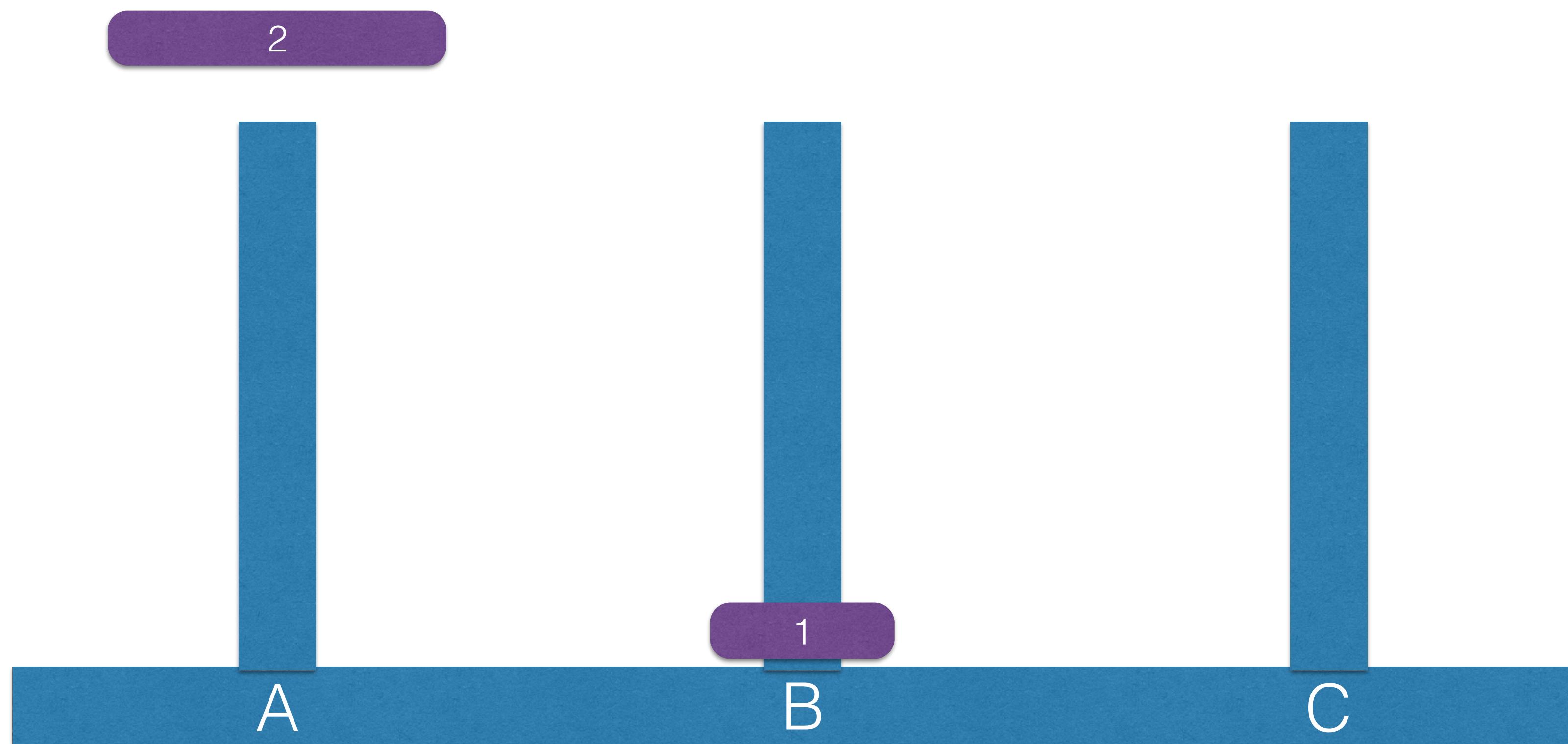
N = 2



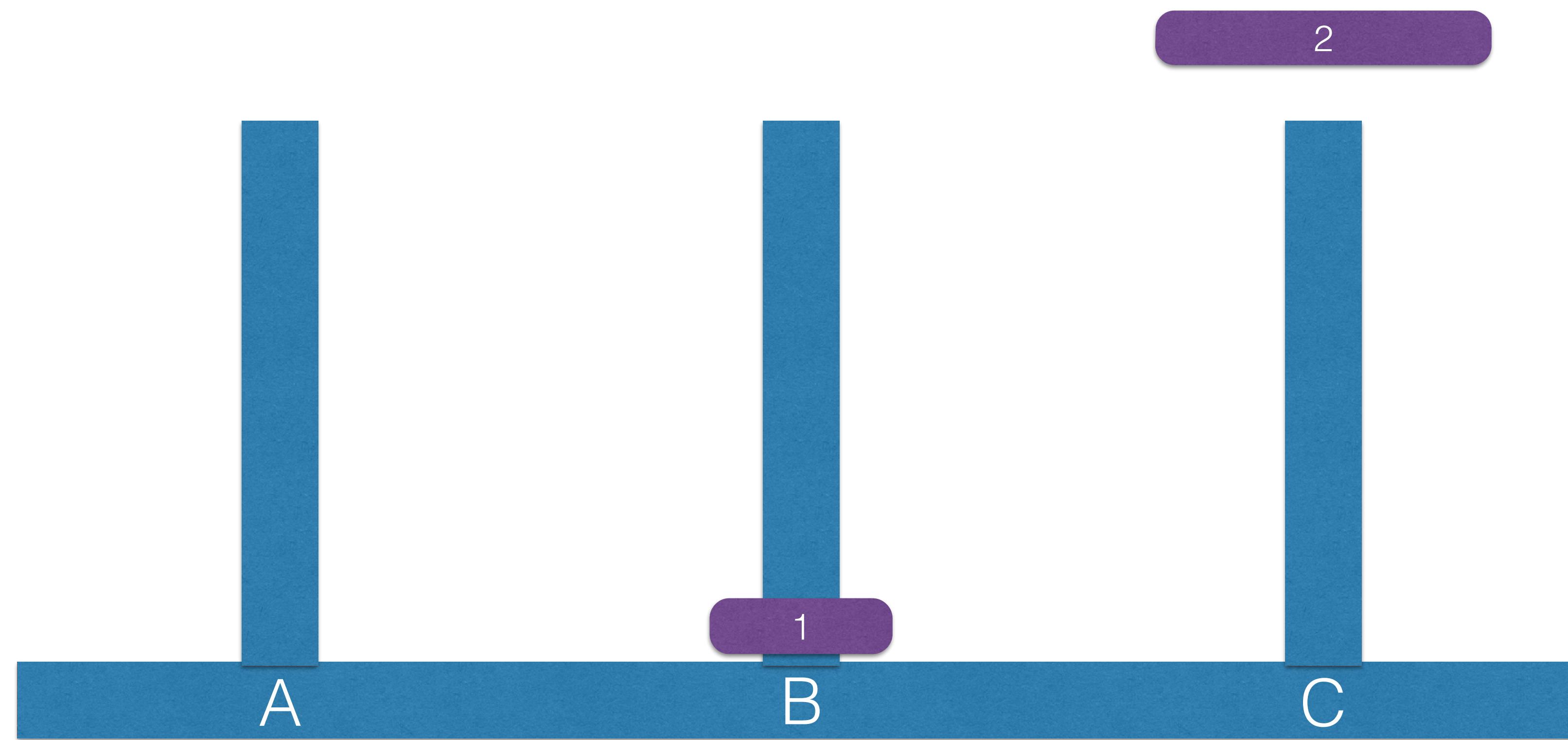
N = 2



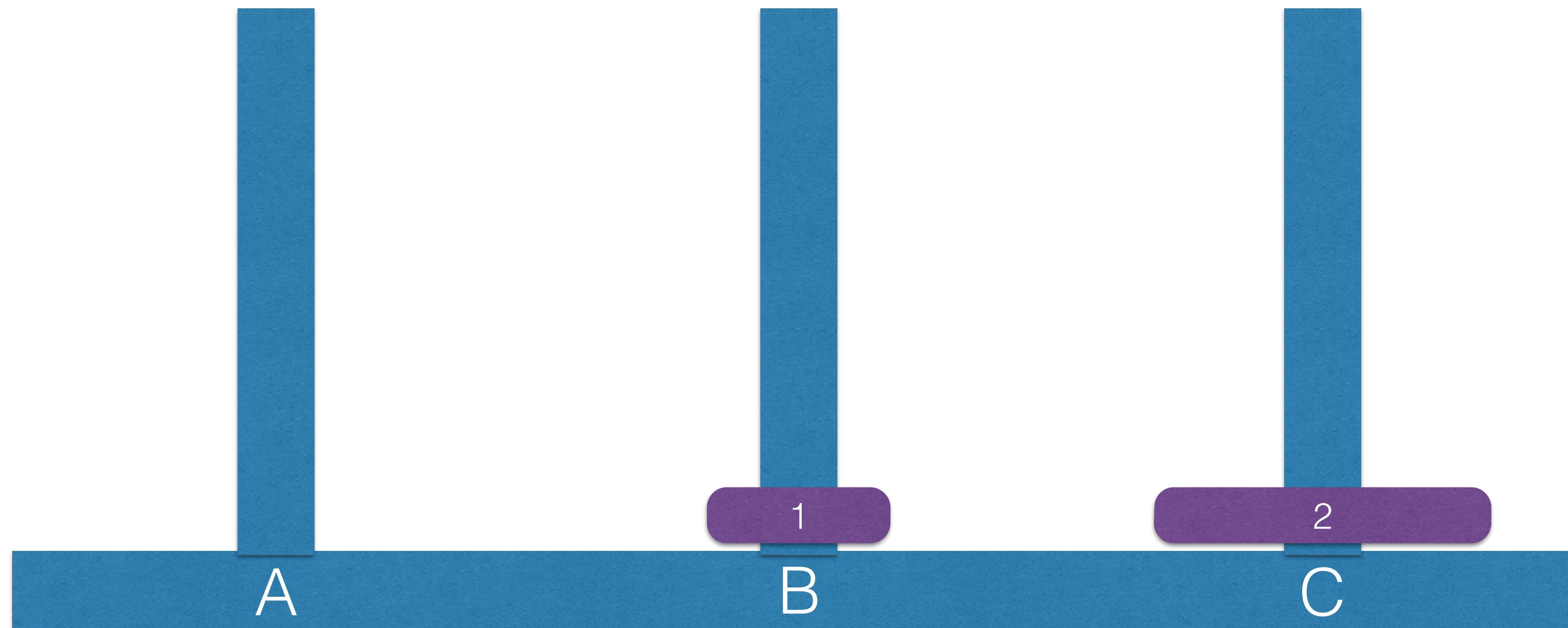
N = 2



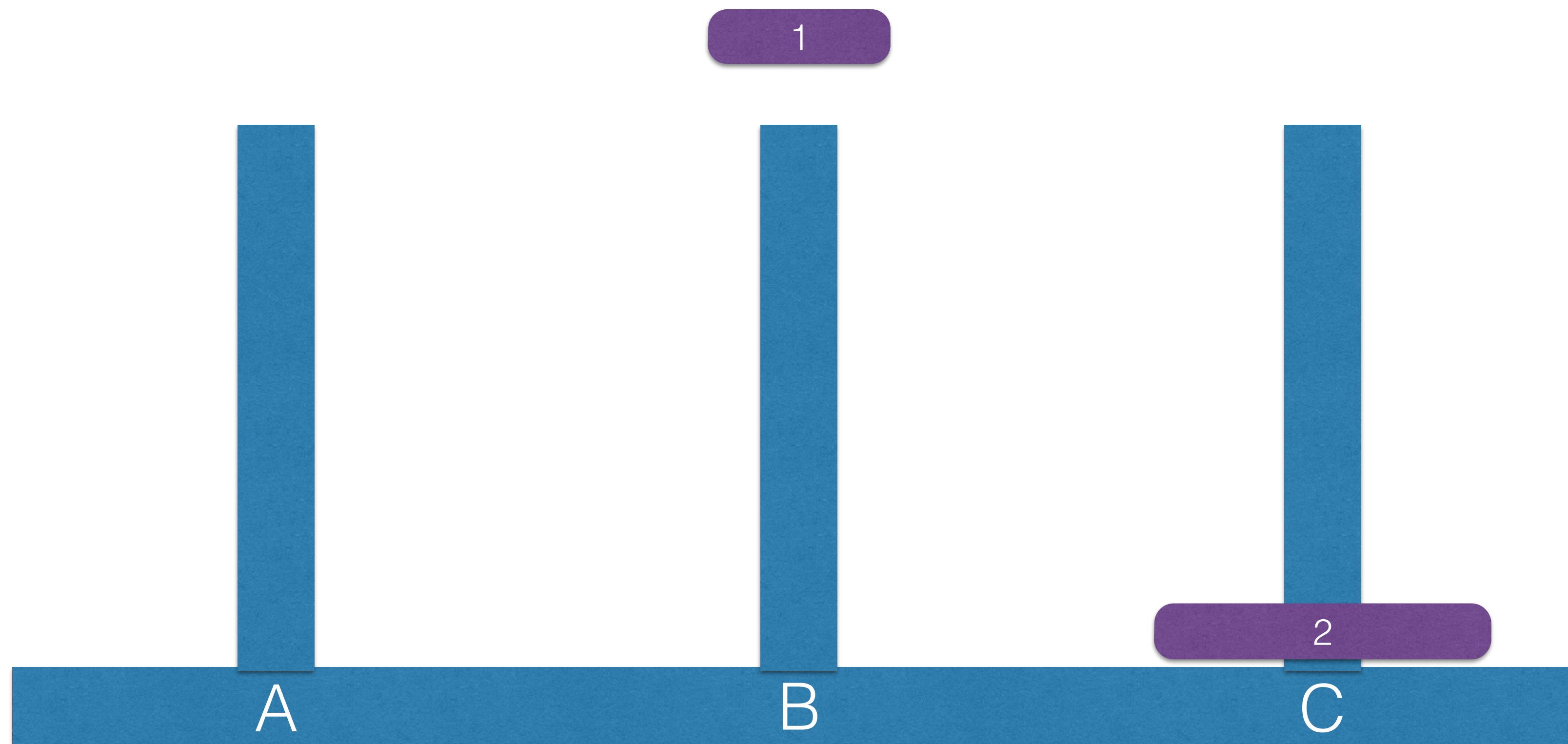
N = 2



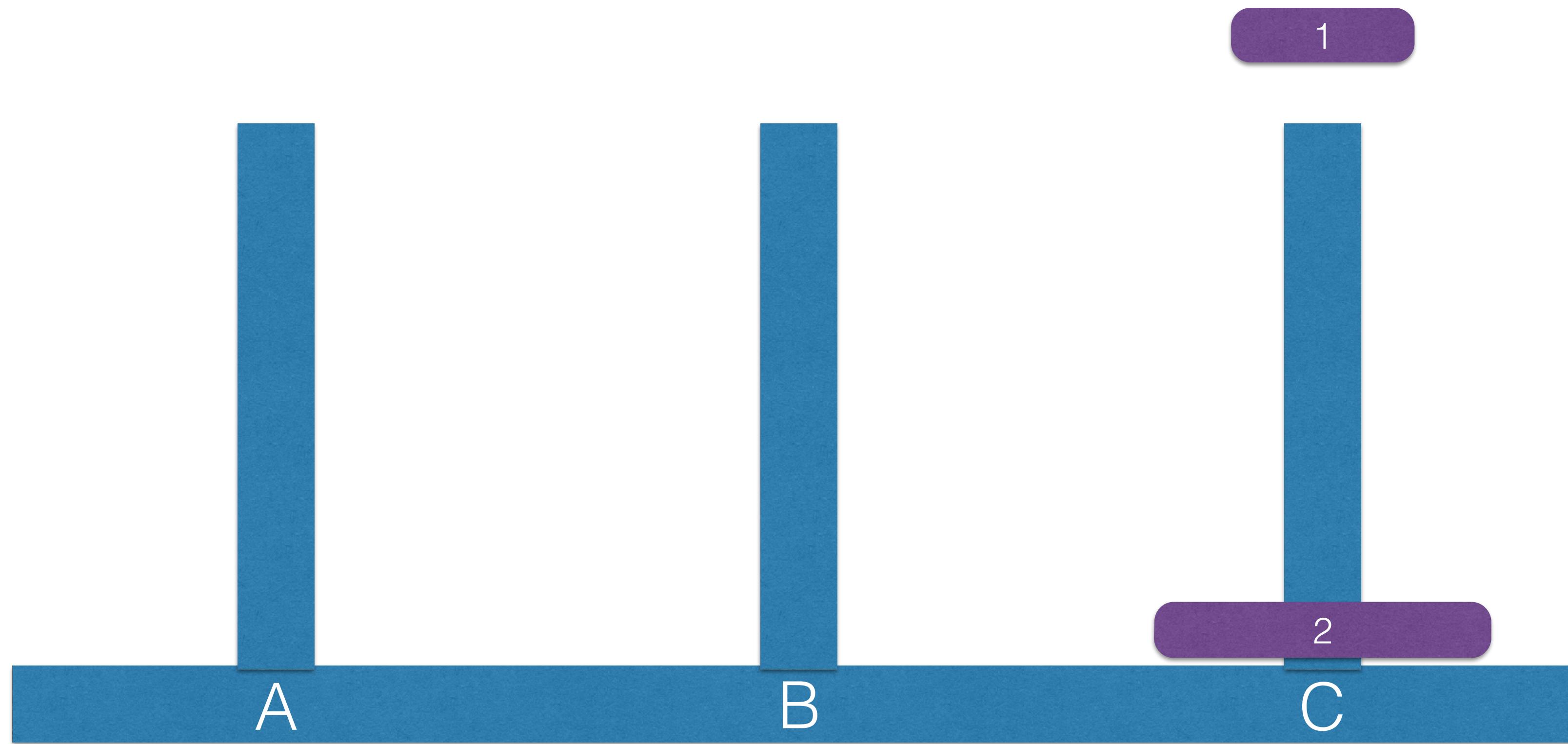
N = 2



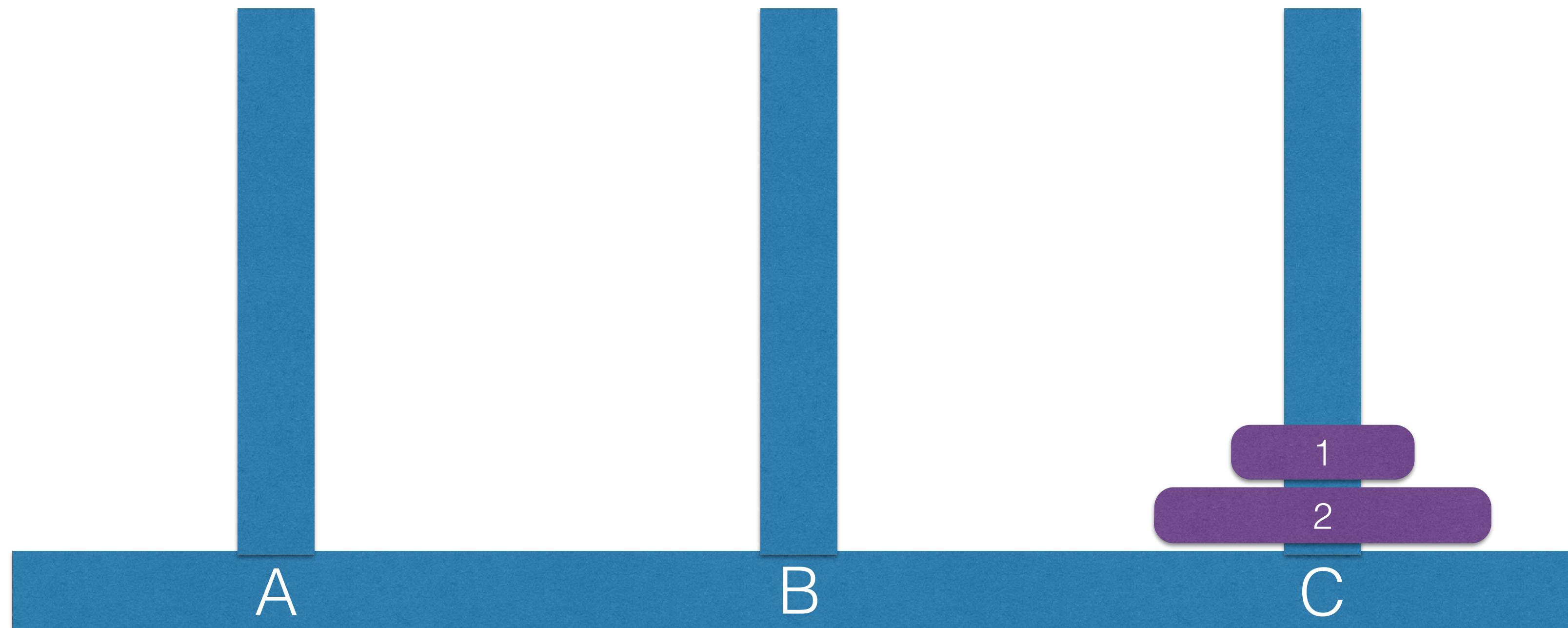
N = 2



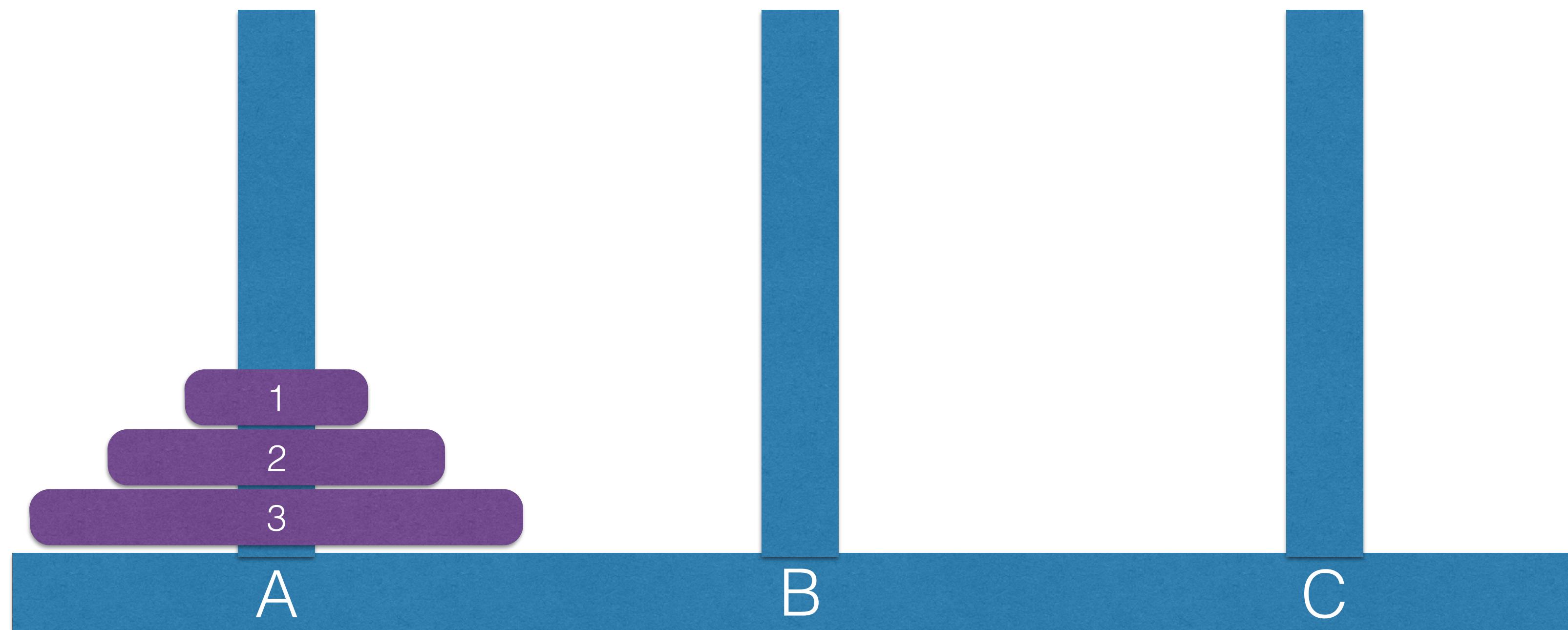
N = 2



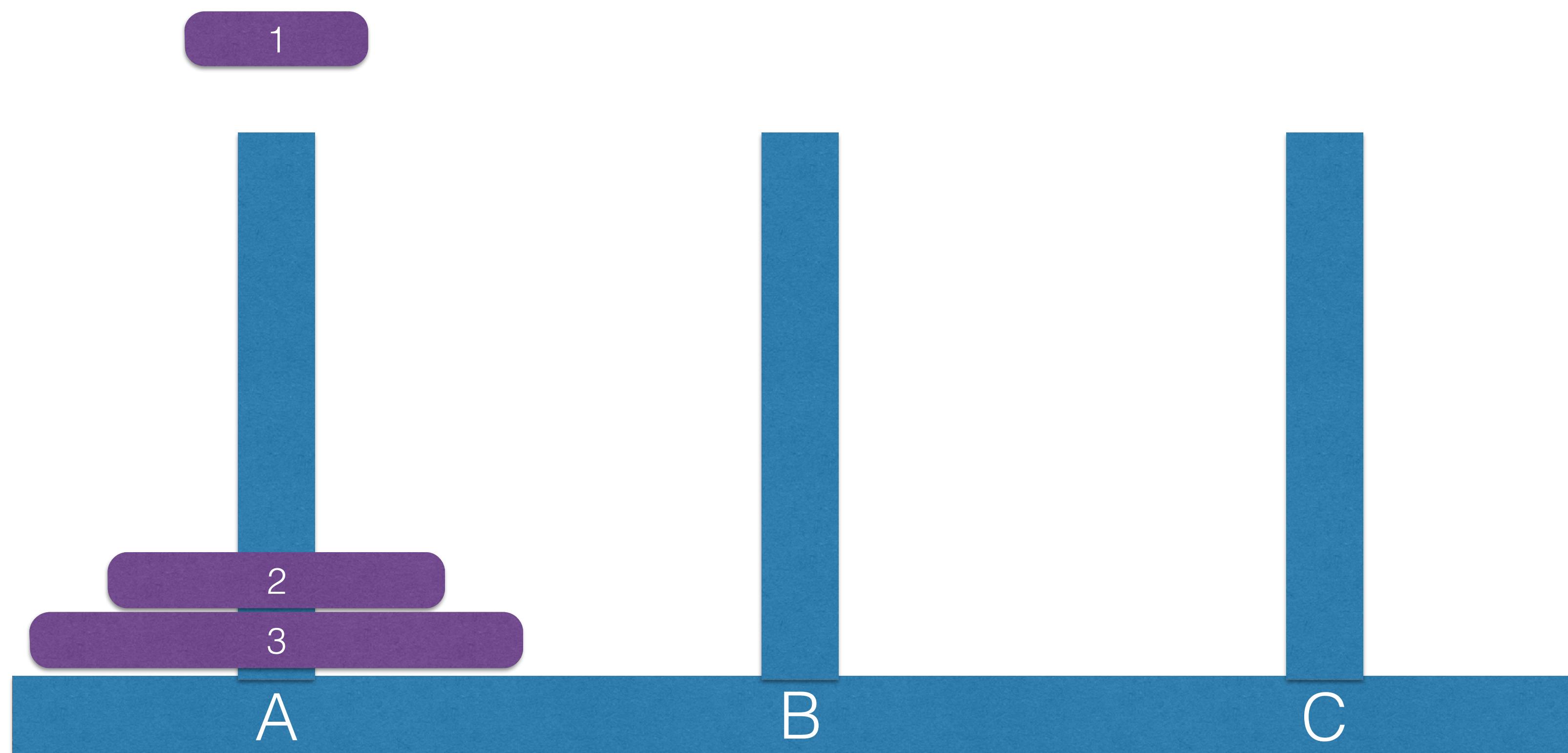
N = 2



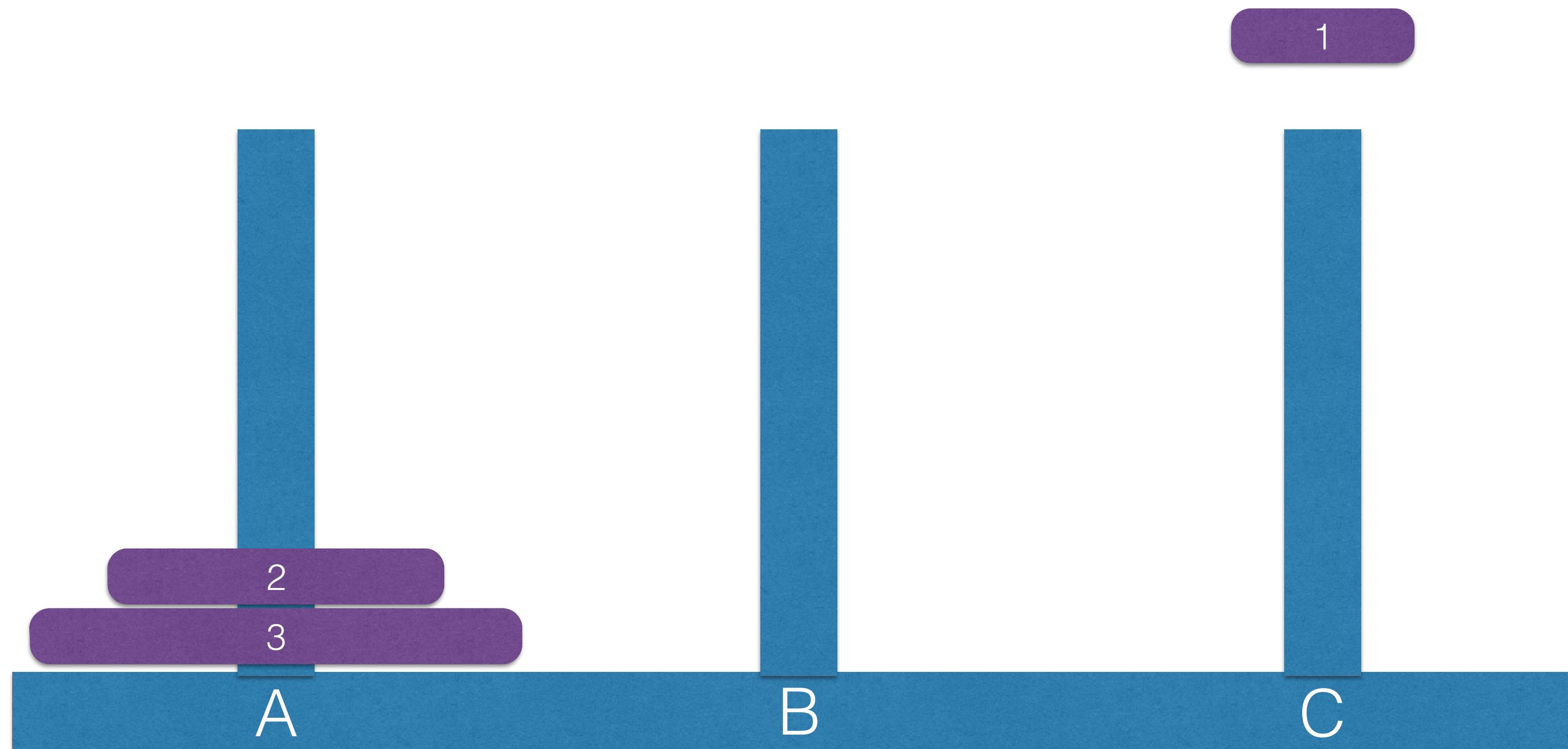
N = 3



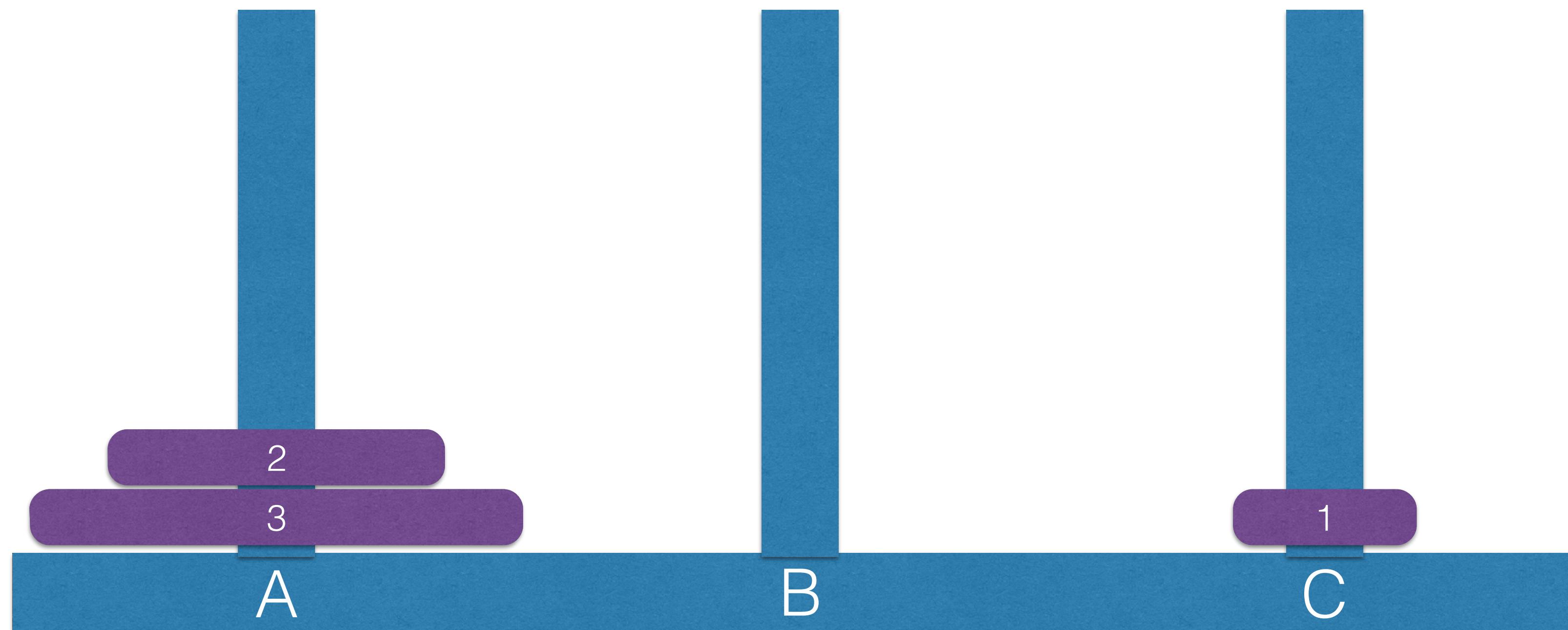
N = 3



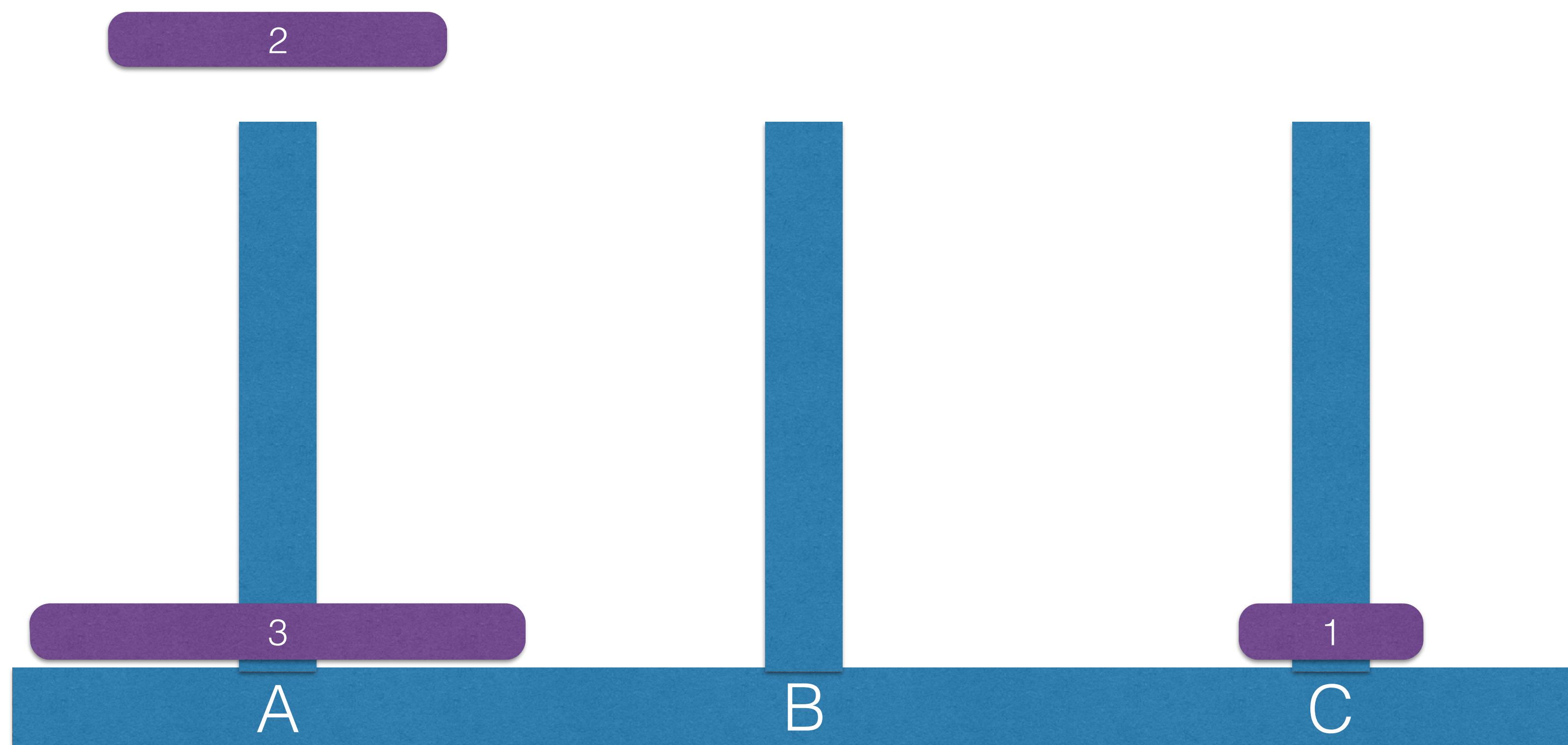
N = 3



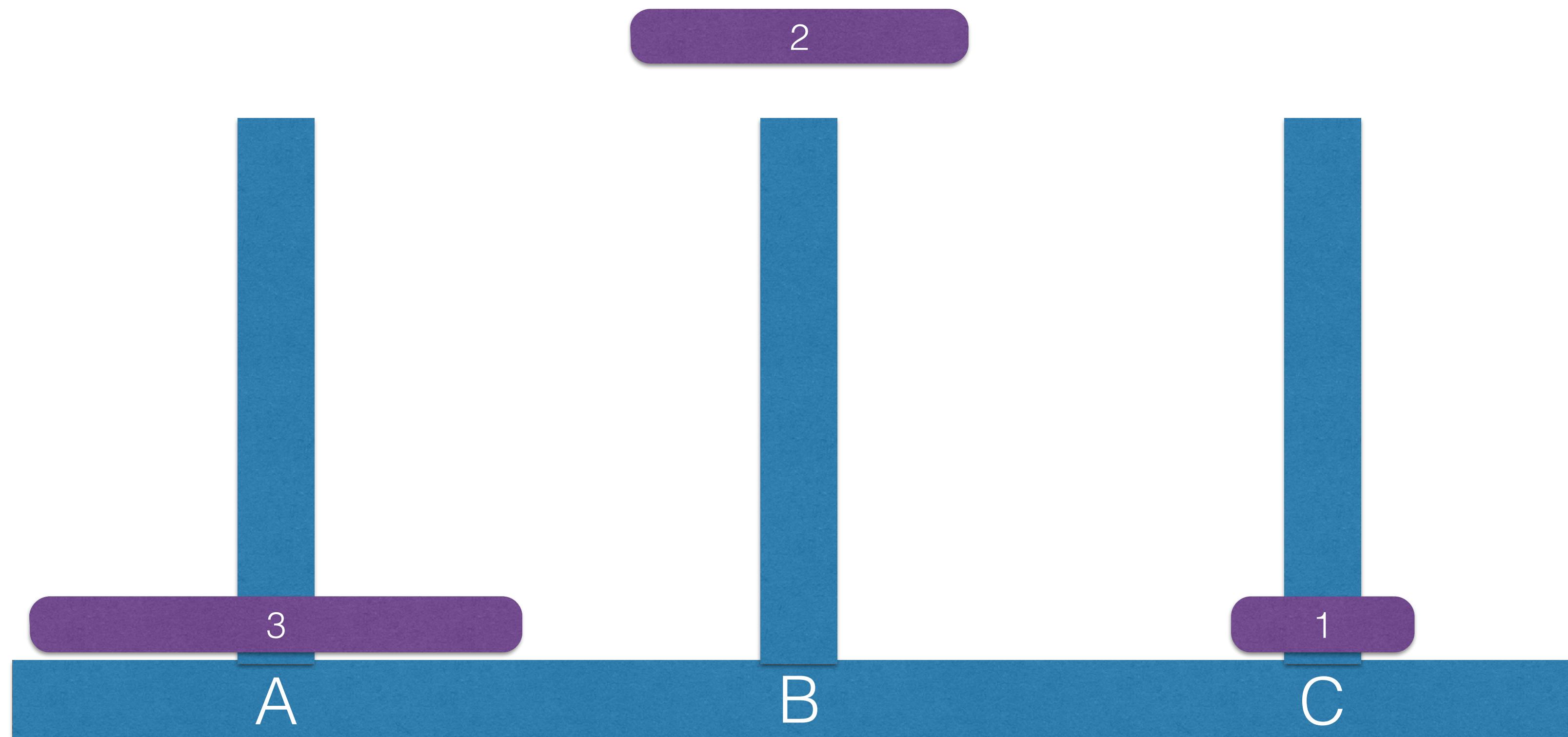
N = 3



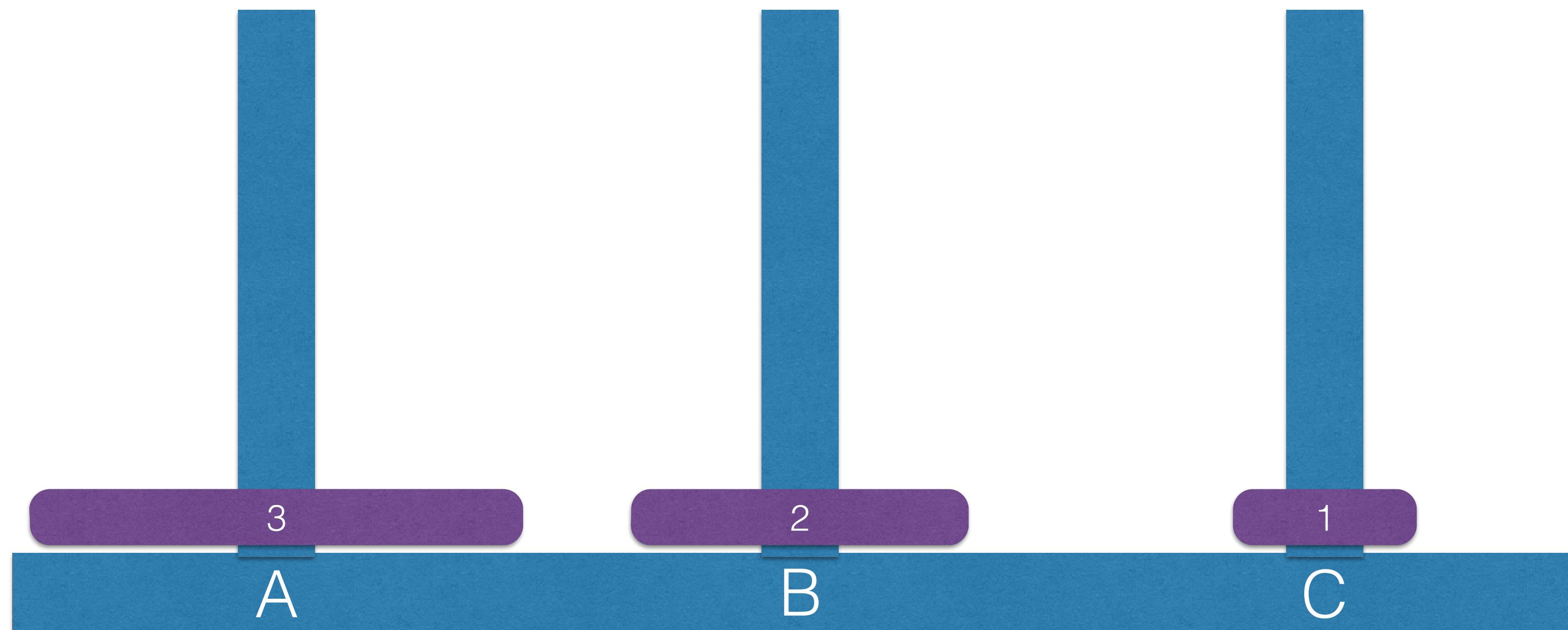
N = 3



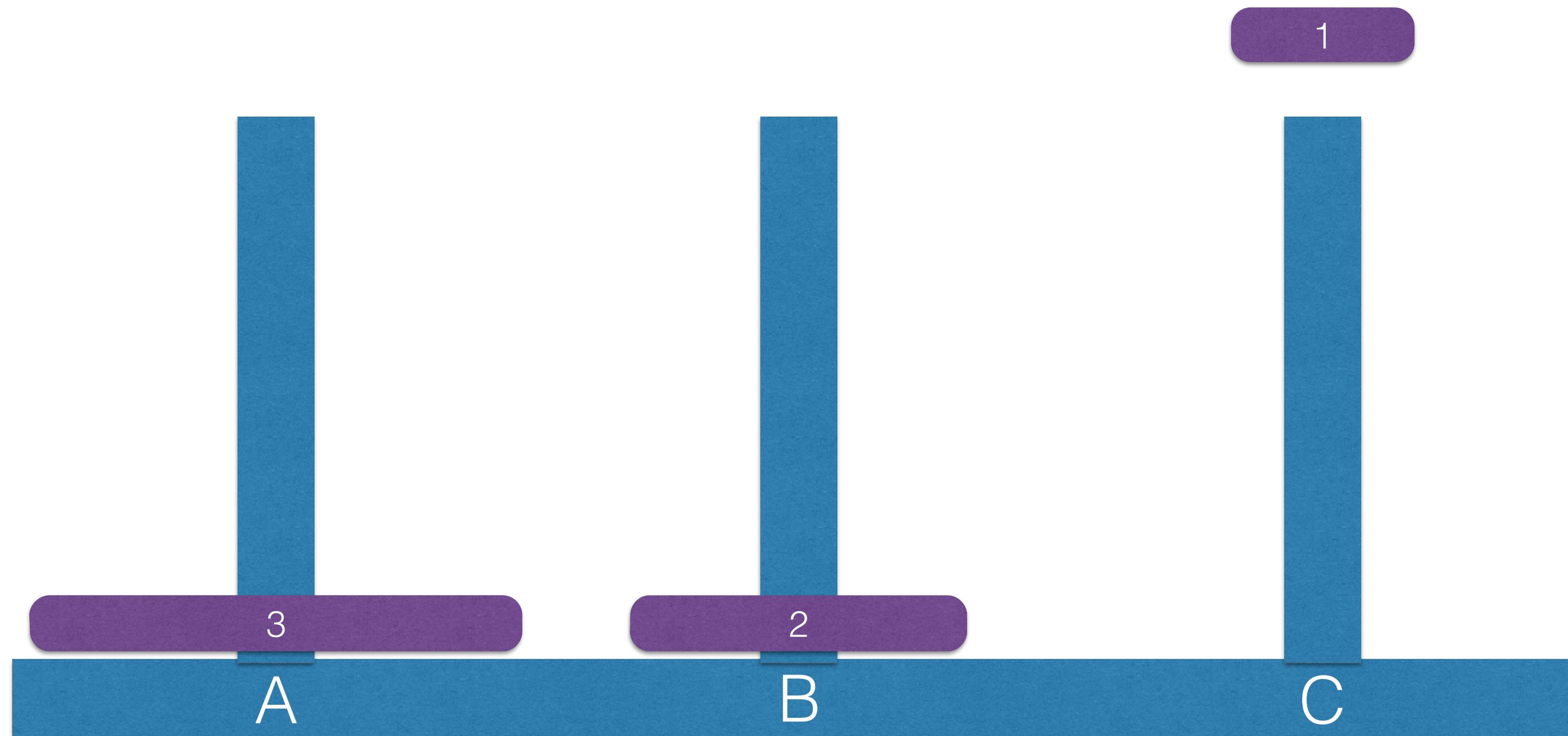
N = 3



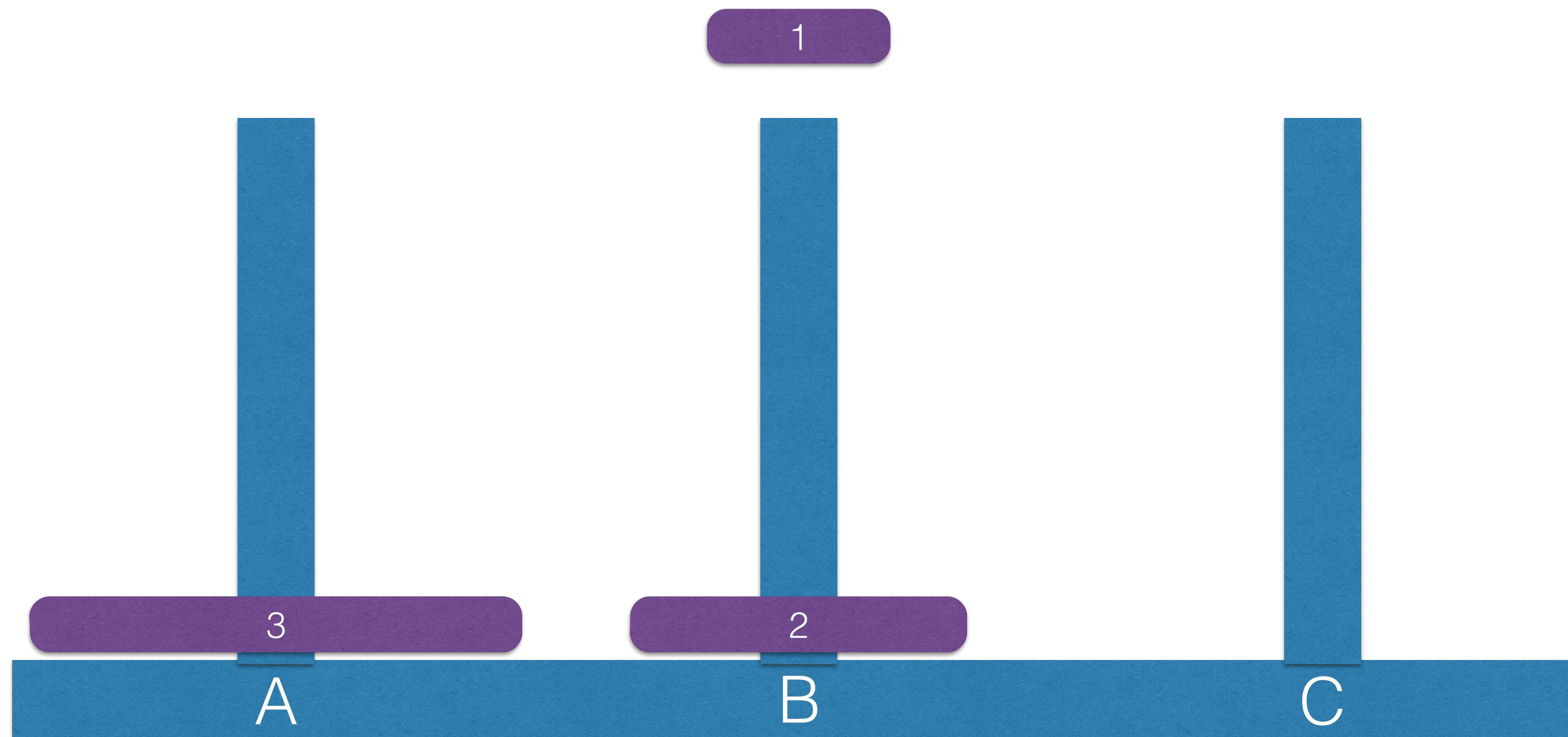
N = 3



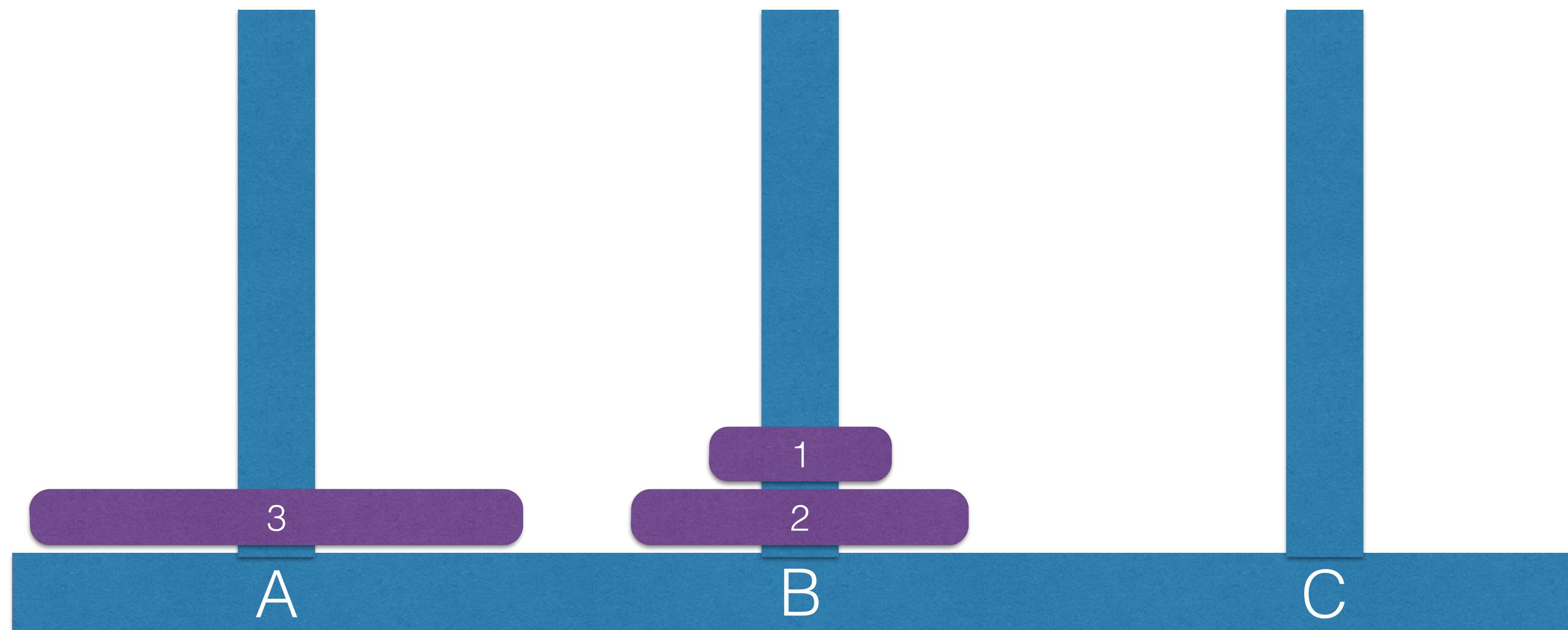
N = 3



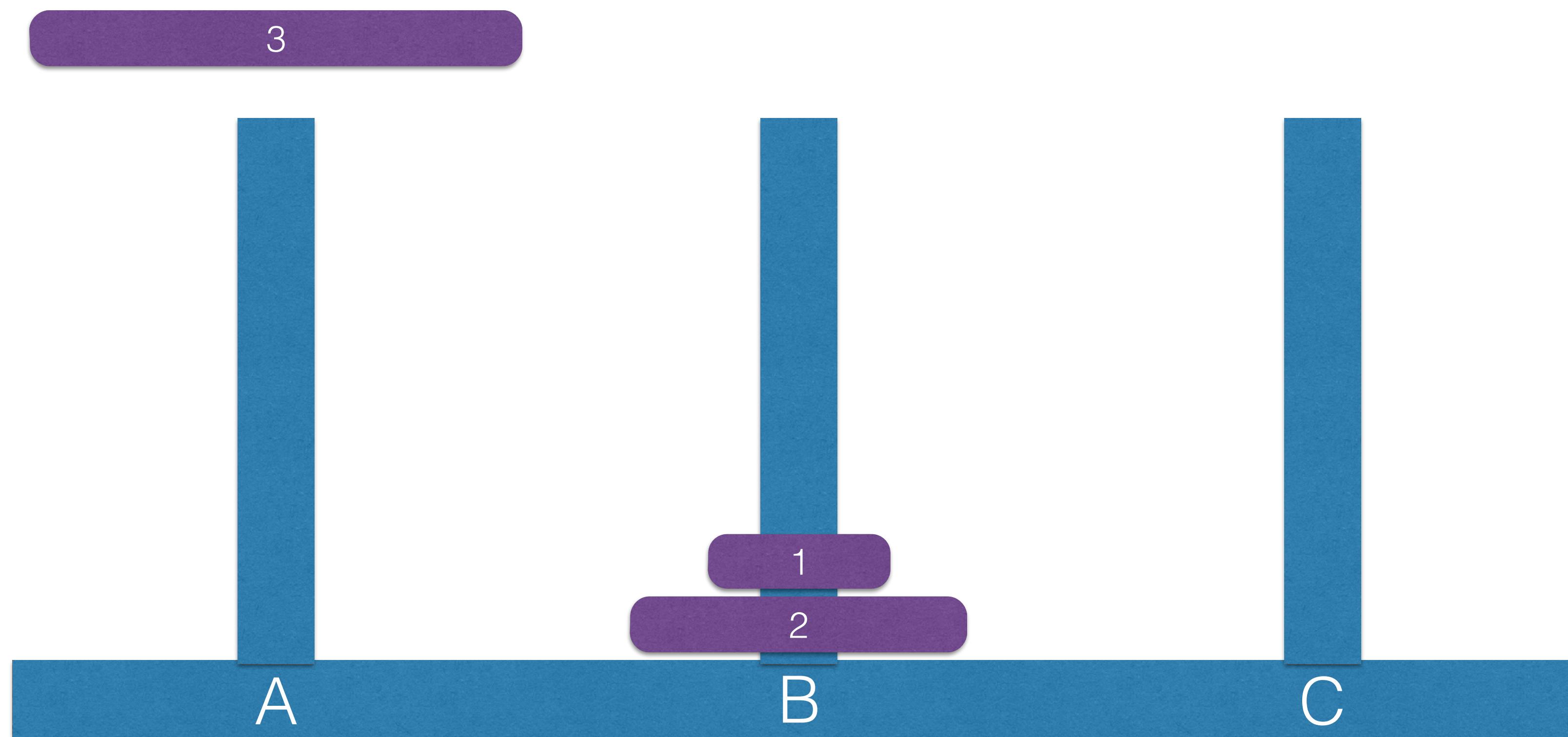
N = 3



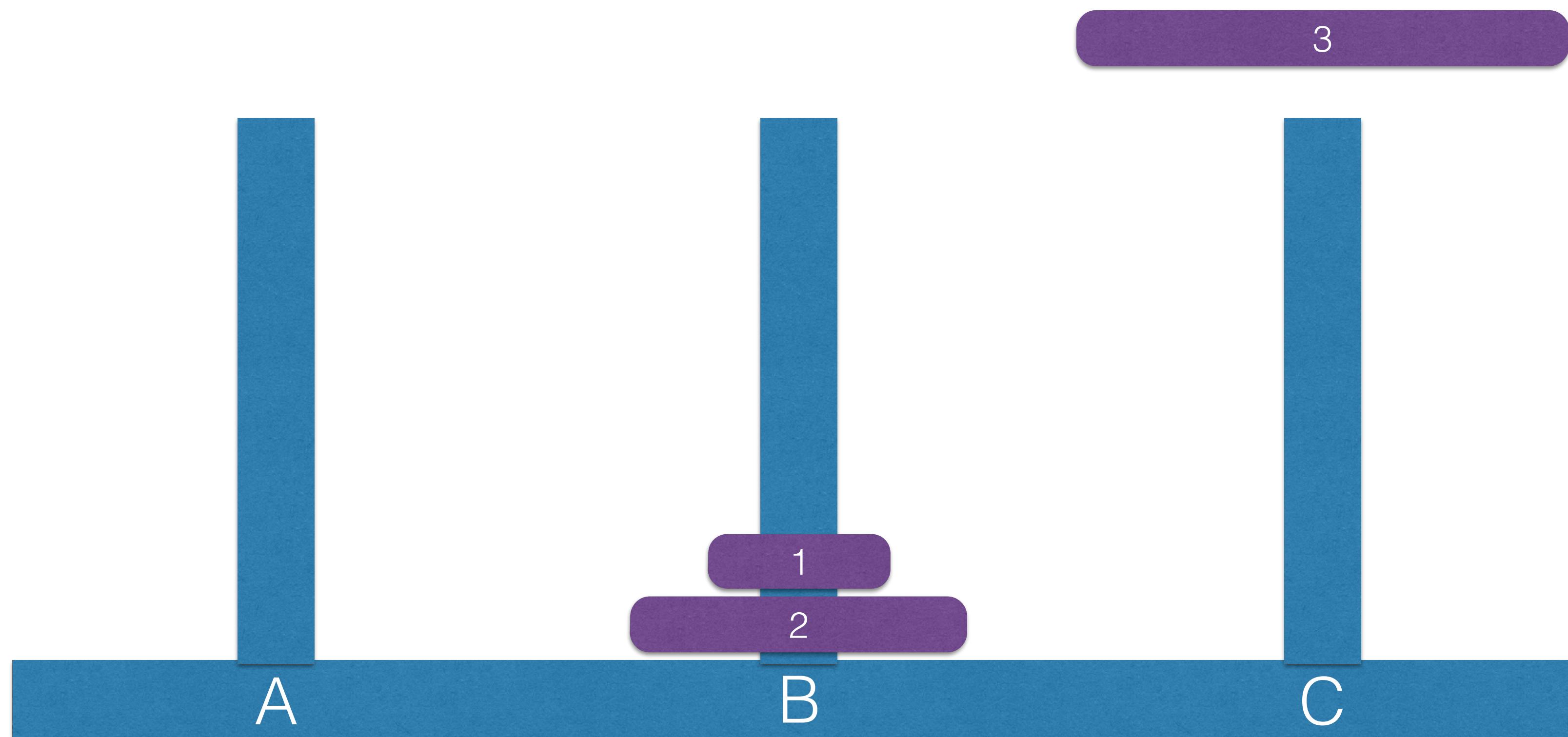
N = 3



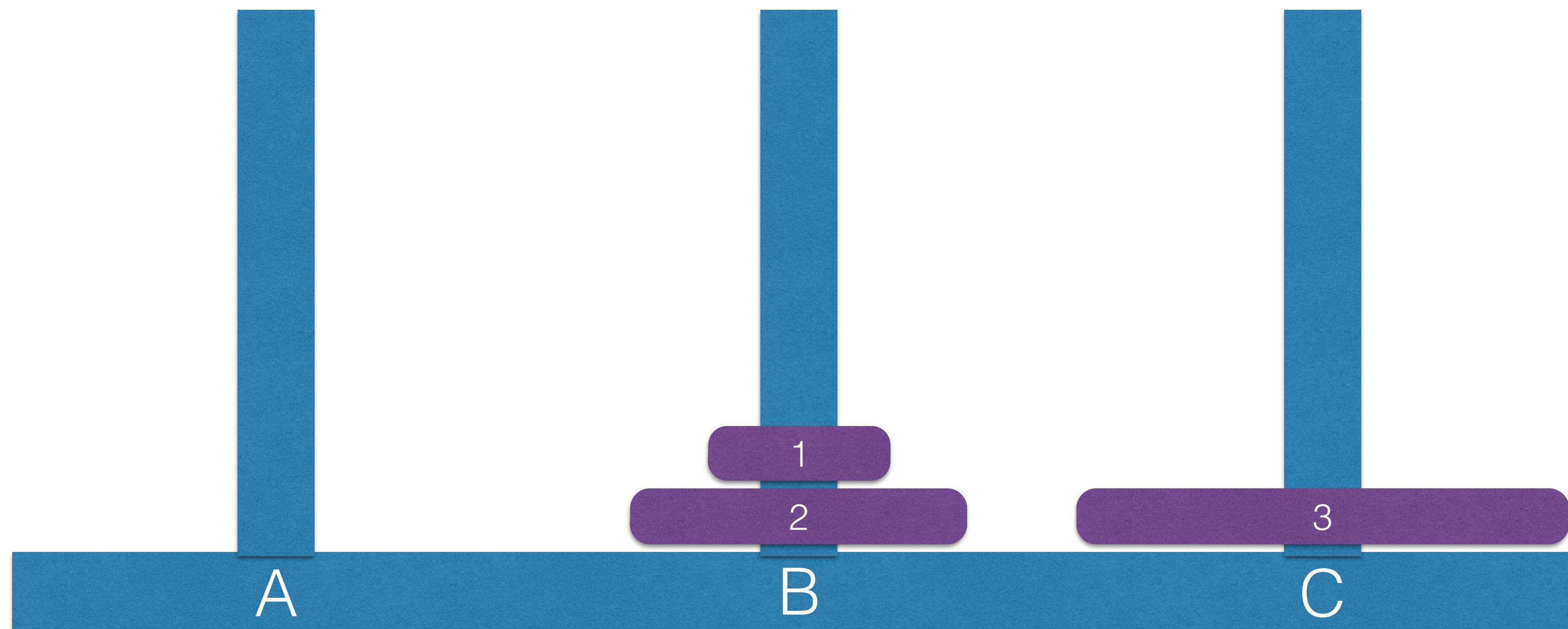
N = 3



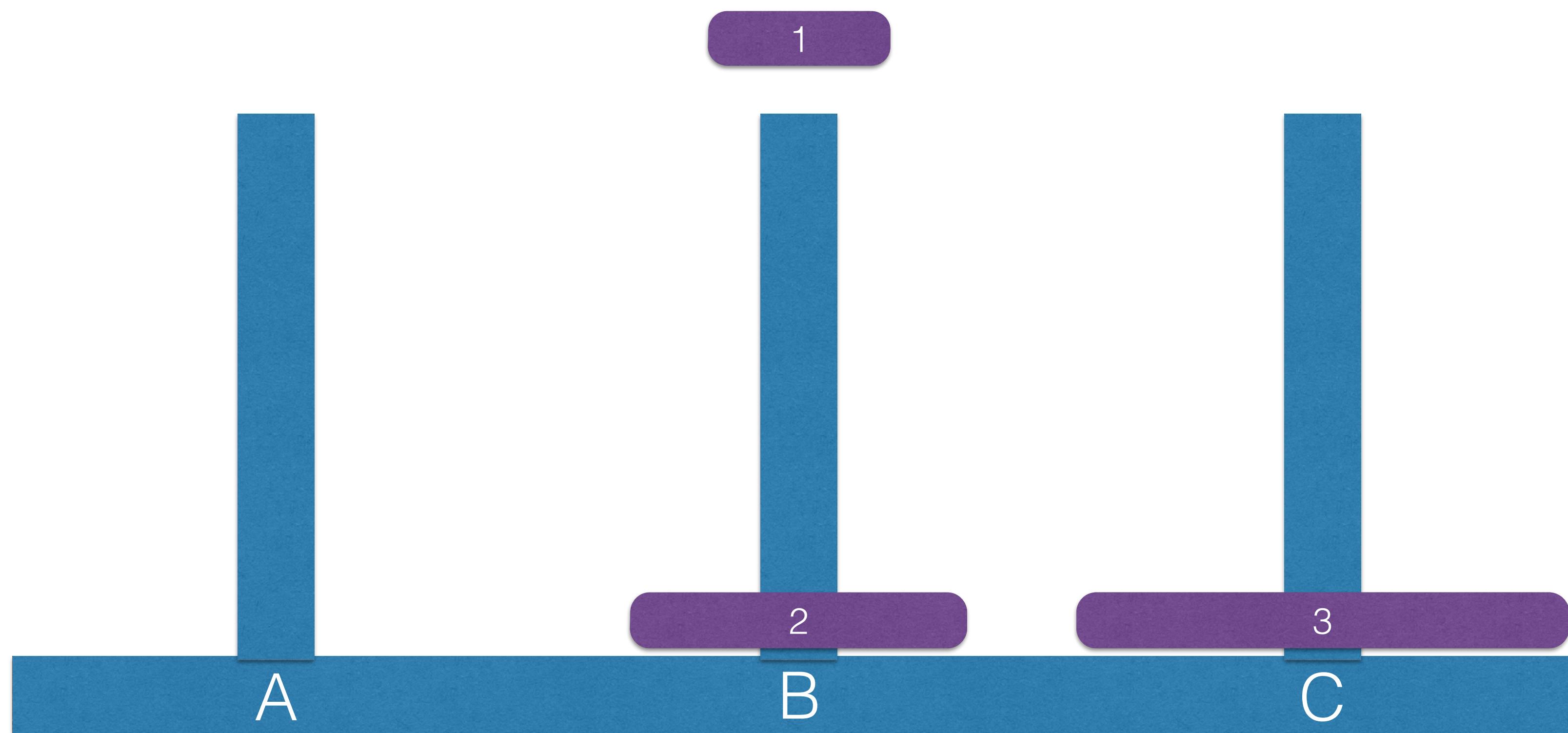
N = 3



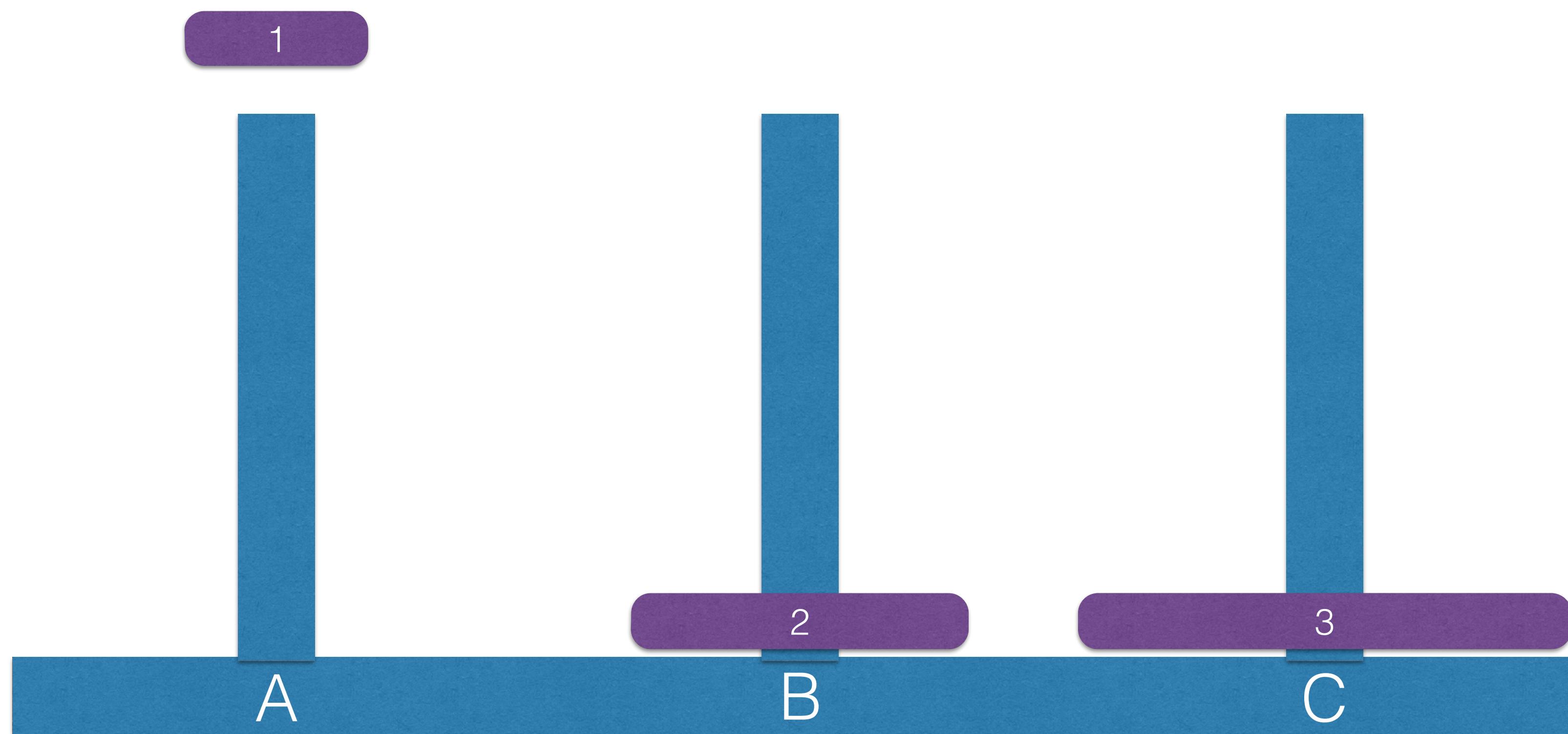
N = 3



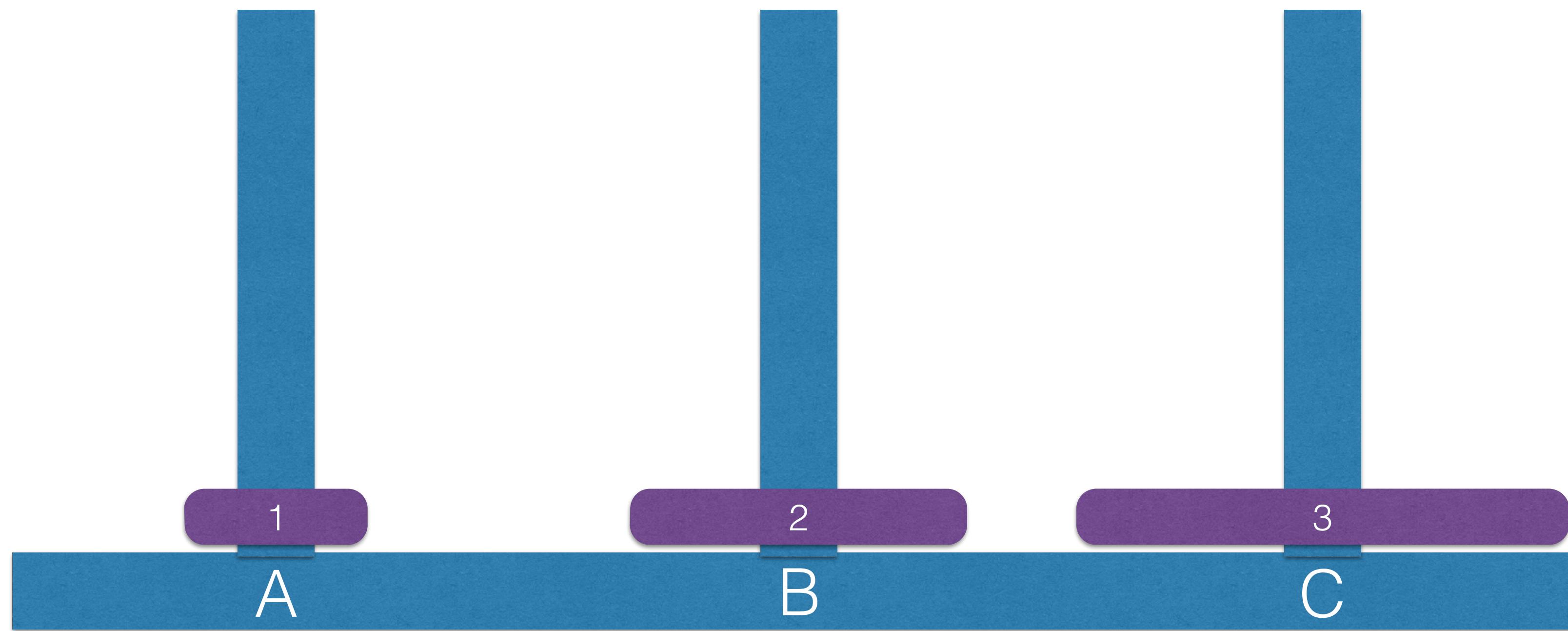
N = 3



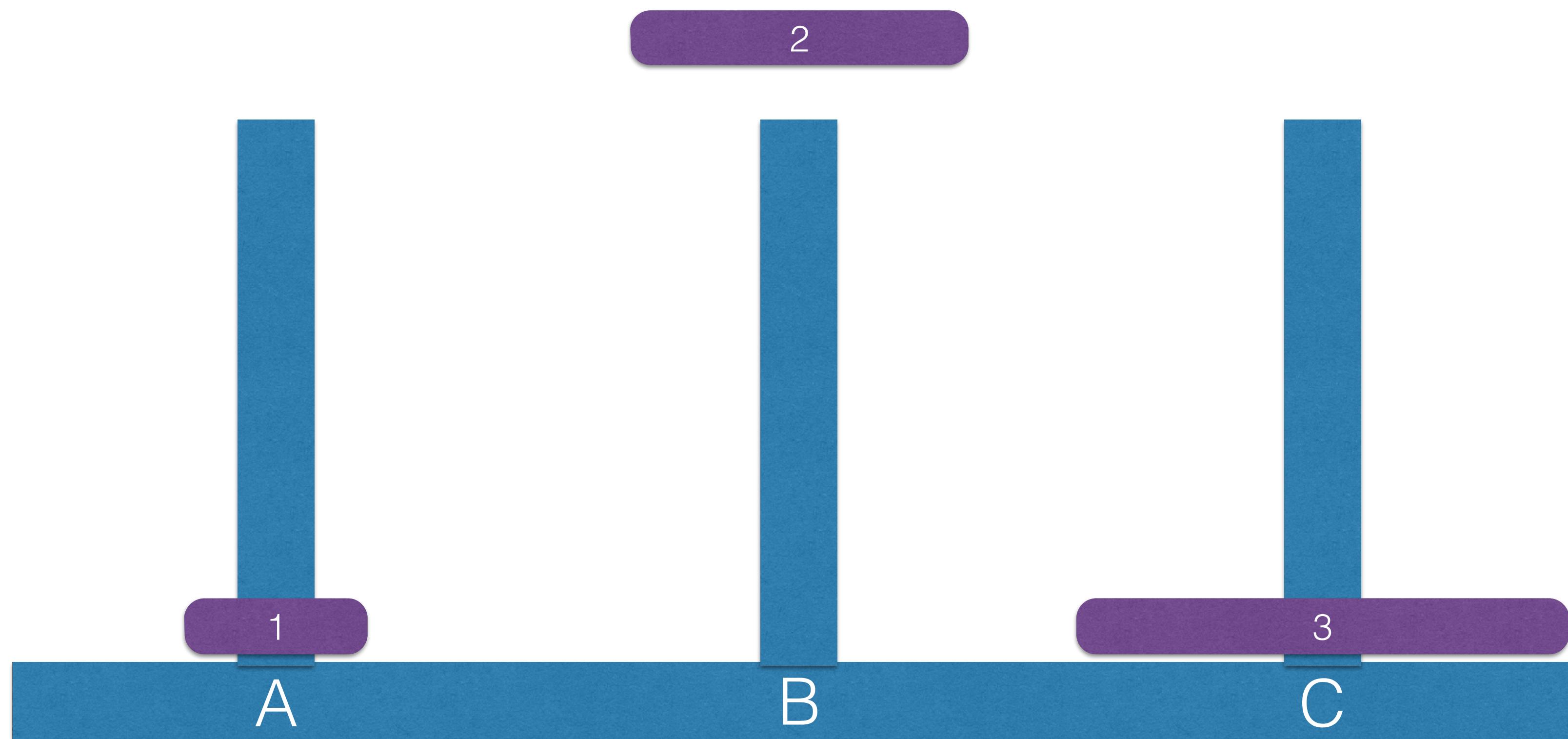
N = 3



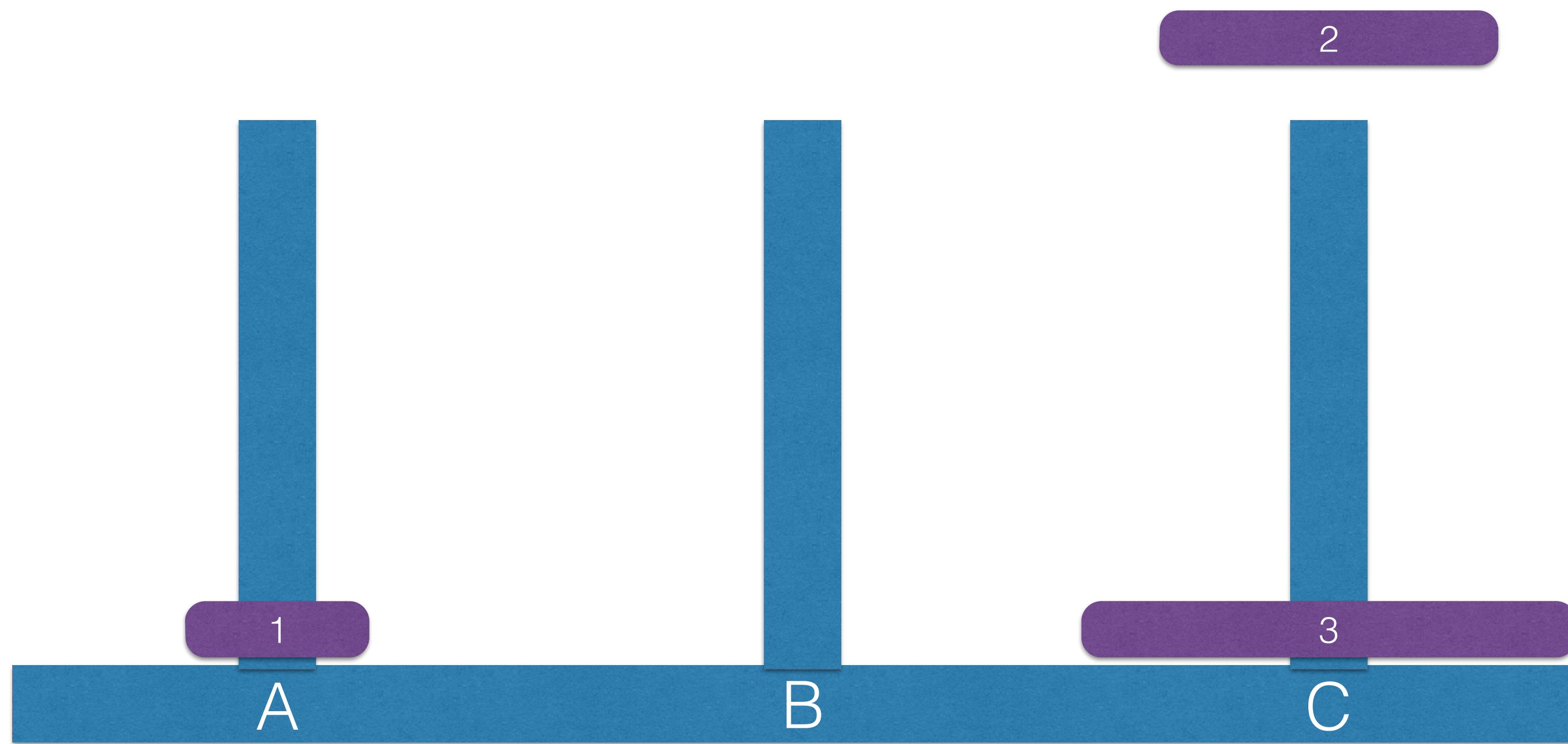
N = 3



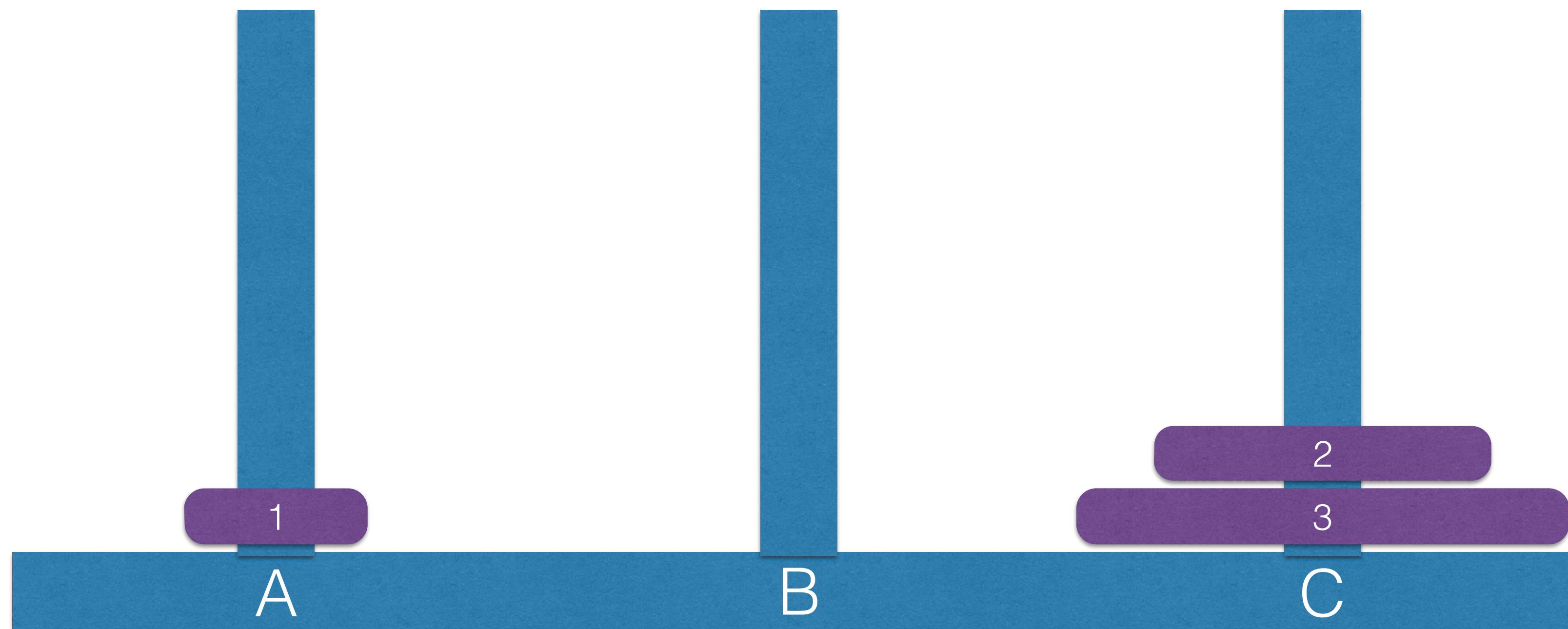
N = 3



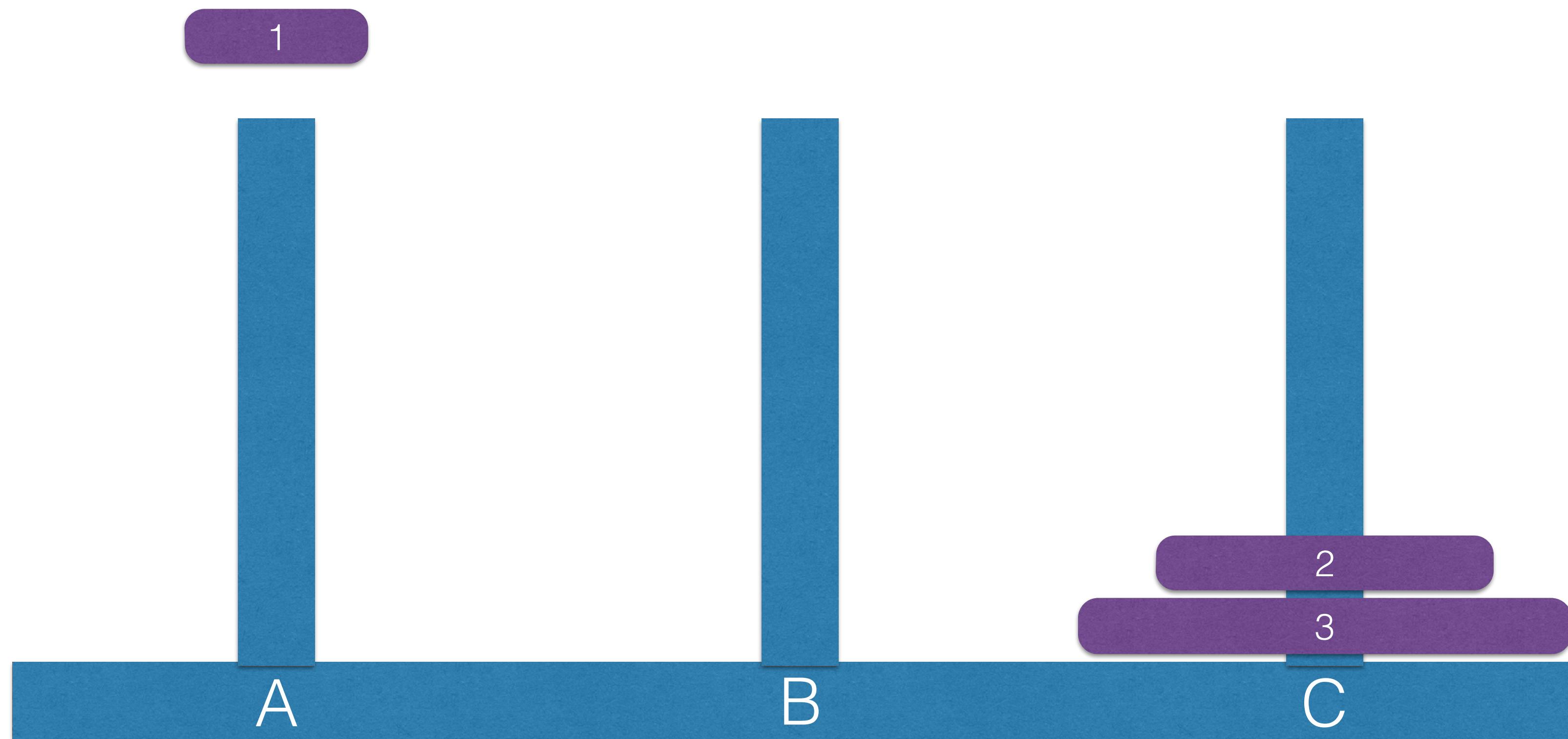
N = 3



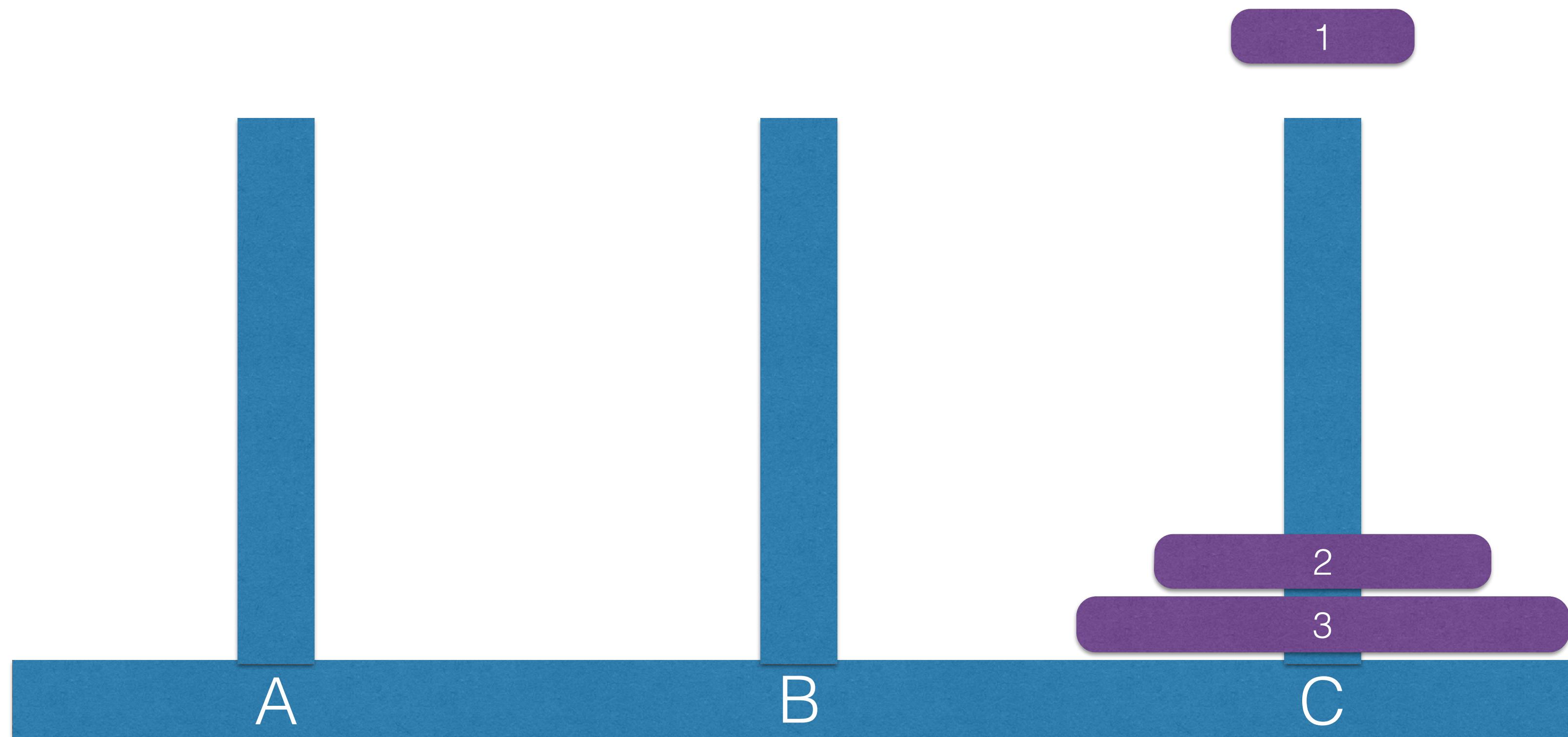
N = 3



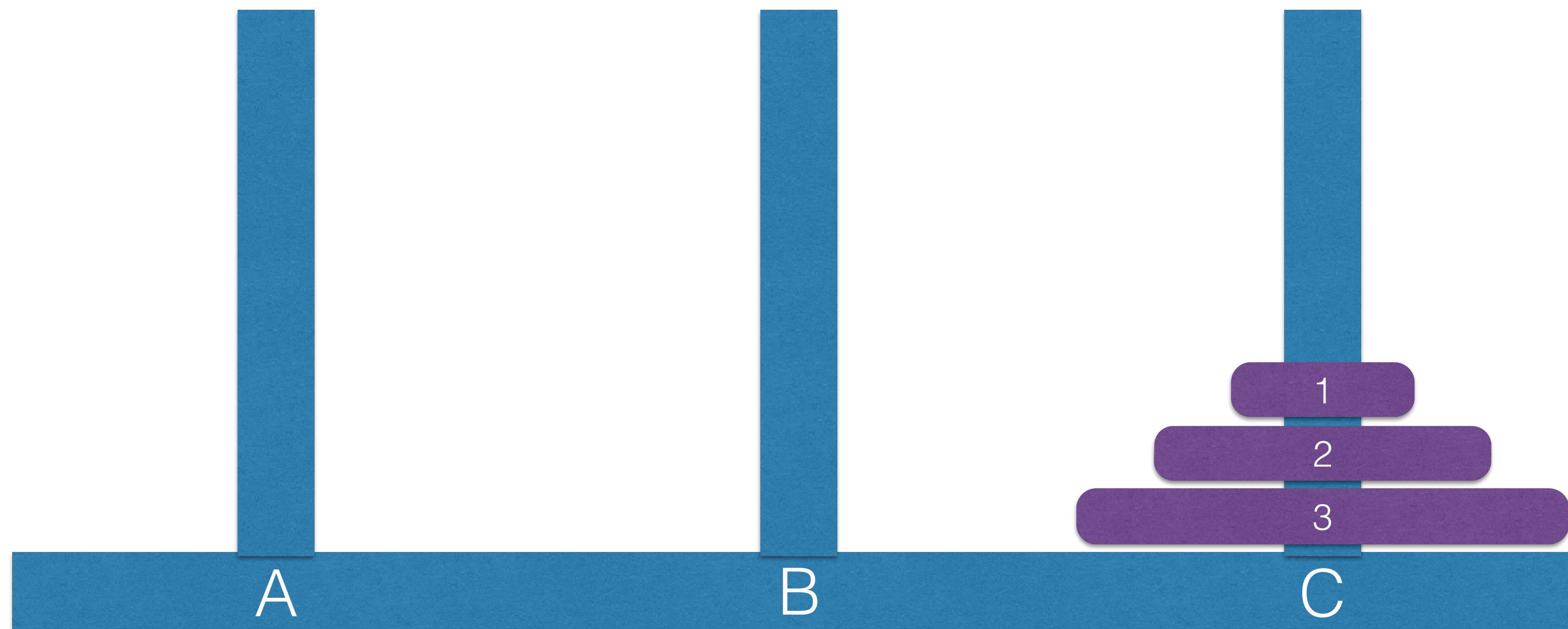
N = 3



N = 3



N = 3



文字版

- $N = 1$
 - A from 1 to 3, done
- $N = 2$
 - A from 1 to 2
 - B from 1 to 3
 - A from 2 to 3, done

文字版

- $N = 3$
- A from 1 to 3
- B from 1 to 2
- A from 3 to 2
- C from 1 to 3
- A from 2 to 1
- B from 2 to 3
- A from 1 to 3, done

發現什麼了嗎？

- 若要把 N 個盤子從 1 搬到 3
 - 需要先把 $N-1$ 個盤子從 1 搬到 2
 - 要把 $N-1$ 個盤子從 1 搬到 2
 - 需要先把 $N-2$ 個盤子從 1 搬到 3
 - ...

換成遞迴表示式

- 盤子只剩一個，把直接從起始柱子搬到終點柱子
- 盤子不只一個：
 - 先把 $N-1$ 個盤子搬到暫放的柱子，把最下面的盤子搬到終點柱子
 - 再把 $N-1$ 個盤子從暫放的柱子搬到終點柱子
- function 可能的模樣：
 - `hanoi(N 個盤子, from 起始柱子, tmp 暫放柱子, to 終點柱子)`