

# 補充：資料結構與演算法

TYIC 桃高資訊社

# 二分搜尋法

二分搜尋法(binary search)是一種常見的搜尋法

在使用二分搜尋法前須將資料排序

因為在搜尋到較目標大的資料時，下次搜尋只會搜尋較小的資料

反之在搜尋到較目標小的資料時，下次搜尋只會搜尋較大的資料

二分搜尋法一次就可以排除一半的可能

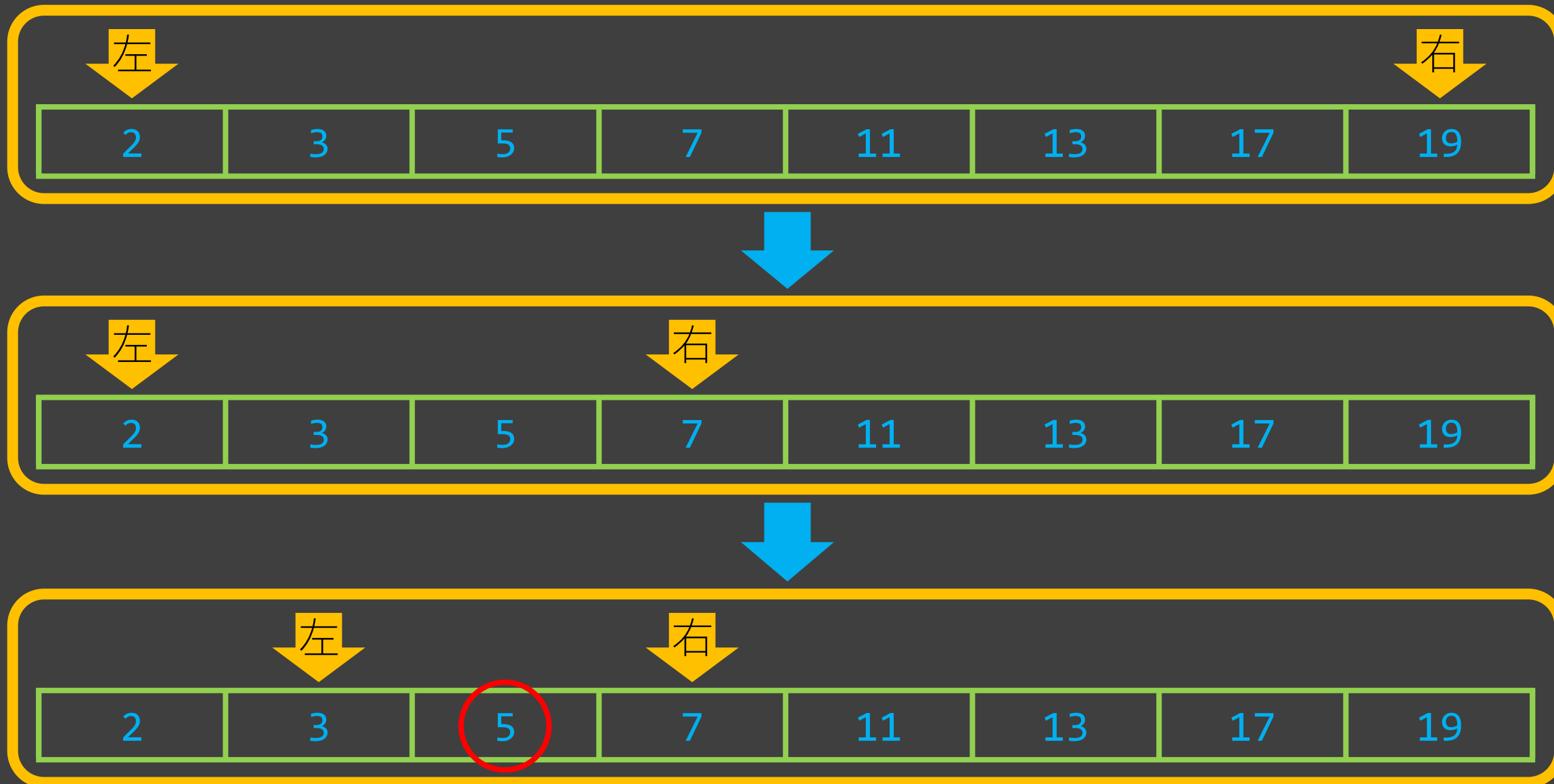
相較於循序搜尋法(線性搜尋法，Linear Search)

依序比對每一個資料直到找到正確的資料

二分搜尋法效率較高，但循序搜尋法的資料不須排序

# 二分搜尋法

找 5



# 二元搜尋樹

二元搜尋樹(binary search tree)是一種特殊的樹(tree)

其效率有時較陣列和鏈結串列較高

樹中的每個資料稱為節點(node)

沒有父節點的資料稱為根節點(root)

沒有子節點的資料稱為葉節點(leave)

某節點的子節點是該某節點的子樹的根節點

從任一葉節點到根節點的最大節點數為該樹的高度(height)

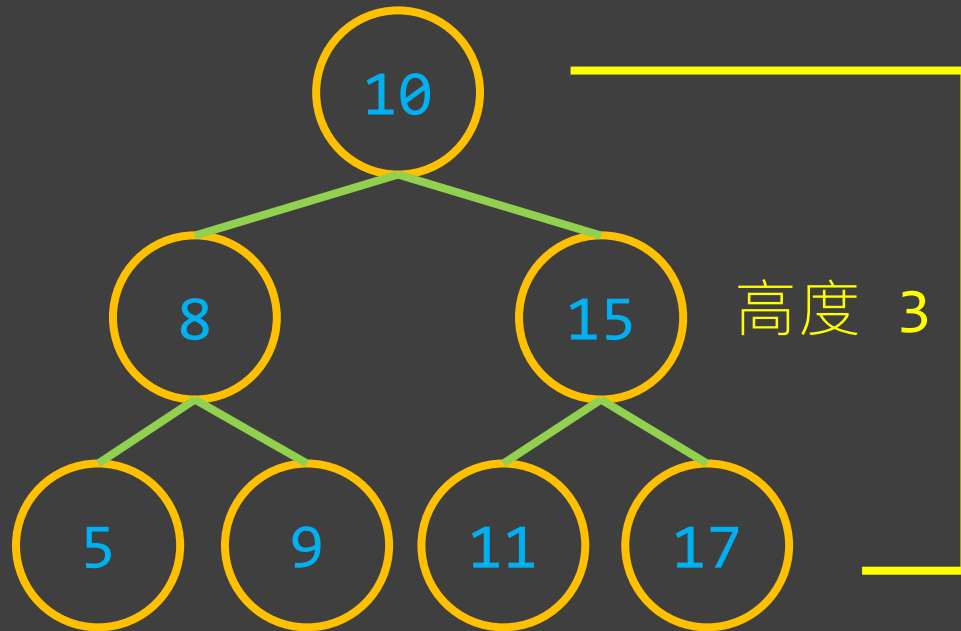
在二元搜尋樹中，比根節點小的資料會放到左子樹中

比根節點大的資料會放到右子樹中

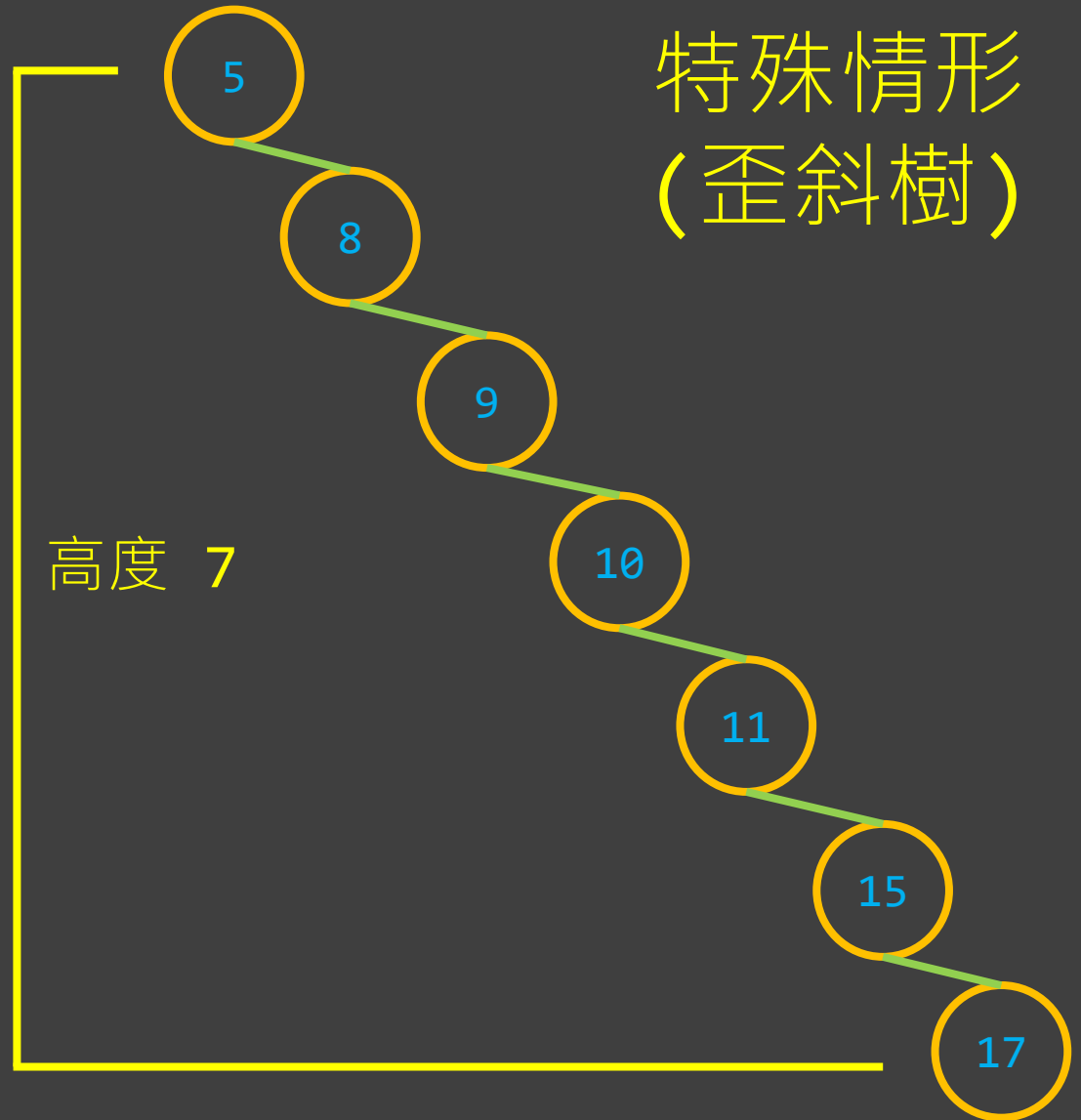
這樣利用二分搜尋法搜尋就會非常快速

# 二元搜尋樹

正常情形



特殊情形  
(歪斜樹)



# 紅黑樹

紅黑樹(**red-black tree**)是一種二元搜尋樹  
且會自平衡(**self-balancing**)，避免出現剛剛的特殊情形  
紅黑樹的葉節點皆為空資料(**null**)，並定義了幾條規則：

1. 節點是紅色或黑色
2. 根節點是黑色
3. 所有的葉節點都是黑色
4. 相接節點不能皆為紅色
5. 從根節點到任一葉節點的黑色節點數量皆相同

這樣的規則使得最長路徑長度不超過最短路徑長度的兩倍

# 紅黑樹

