

เฉลยแบบฝึกหัด Divide and Conquer

1. จงแสดงขั้นตอนการหาแก้ปัญหาหานี้ โดยใช้แนวคิดแบบ Divide and Conquer

- $2^{1000} \bmod 10$
 $2^{1000} \bmod 10 = (2^{500} \bmod 10)^2 \bmod 10 = 6$
 $2^{500} \bmod 10 = (2^{250} \bmod 10)^2 \bmod 10 = 6$
 $2^{250} \bmod 10 = (2^{125} \bmod 10)^2 \bmod 10 = 4$
 $2^{125} \bmod 10 = 2 * (2^{62} \bmod 10)^2 \bmod 10 = 2$
 $2^{62} \bmod 10 = (2^{31} \bmod 10)^2 \bmod 10 = 4$
 $2^{31} \bmod 10 = 2 * (2^{15} \bmod 10)^2 \bmod 10 = 8$
 $2^{15} \bmod 10 = 2 * (2^7 \bmod 10)^2 \bmod 10 = 8$
 $2^7 \bmod 10 = 2 * (2^3 \bmod 10)^2 \bmod 10 = 8$
 $2^3 \bmod 10 = 2 * (2^1 \bmod 10)^2 \bmod 10 = 8$
 $2^1 \bmod 10 = 2 * (2^0 \bmod 10)^2 \bmod 10 = 2$
 $2^0 \bmod 10 = 1$
คำตอบ 6

- $3^{600} \bmod 5$
 $3^{600} \bmod 5 = (3^{300} \bmod 5)^2 \bmod 5 = 1$
 $3^{300} \bmod 5 = (3^{150} \bmod 5)^2 \bmod 5 = 1$
 $3^{150} \bmod 5 = (3^{75} \bmod 5)^2 \bmod 5 = 4$
 $3^{75} \bmod 5 = 3 * (3^{37} \bmod 5)^2 \bmod 5 = 2$
 $3^{37} \bmod 5 = 3 * (3^{18} \bmod 5)^2 \bmod 5 = 3$
 $3^{18} \bmod 5 = (3^9 \bmod 5)^2 \bmod 5 = 4$
 $3^9 \bmod 5 = 3 * (3^4 \bmod 5)^2 \bmod 5 = 3$
 $3^4 \bmod 5 = (3^2 \bmod 5)^2 \bmod 5 = 1$
 $3^2 \bmod 5 = (3^1 \bmod 5)^2 \bmod 5 = 4$
 $3^1 \bmod 5 = 3 * (3^0 \bmod 5)^2 \bmod 5 = 2$
 $3^0 \bmod 5 = 3$
คำตอบ 1

2. Binary Search จัดเป็น Divide and conquer algorithm หรือไม่ เพราะเหตุใด

- ไม่ใช่ แม้ว่าแนวคิดของ Binary Search ดูเหมือนว่าจะเป็น Divide and conquer algorithm คือ มีการแบ่งย่อยปัญหาแล้วค่อยๆ แก้ แต่การทำ Binary Search ยังขาดกระบวนการในการรวมคำตอบของปัญหาย่อยๆ โดยทำการแบ่งครึ่งข้อมูลเพื่อหาข้อมูลที่ต้องการเท่านั้น
- Binary Search จัดเป็น Decrease and conquer algorithm คือ พยายามลดปัญหา เพื่อให้แก้ปัญหได้ง่ายขึ้น

3. จงกำหนด base case และ recursive case ของอัลกอริทึม binary search

- $Bs(list, target, begin, end)$ หมายถึง การทำ Binary Search เพื่อค้นหาข้อมูล target จากवेข้อมูล list โดยตำแหน่งเริ่มต้นและสิ้นสุดของ list คือ begin และ end ตามลำดับ
- $Bs(list, target, begin, end)$ จะคืนค่าตำแหน่งของข้อมูล target ที่เราต้องการ

- Base case :
 - ☐ กรณีที่ $\text{target} == \text{list}[\text{mid}]$ จะ return mid
 - ☐ กรณีที่ $\text{begin} > \text{end}$ จะ return -1
- Recursive case :
 - ☐ กรณีที่ $\text{target} > \text{list}[\text{mid}]$ จะเรียก $\text{Bs}(\text{list}, \text{target}, \text{mid}+1, \text{end})$
 - ☐ กรณีที่ $\text{target} < \text{list}[\text{mid}]$ จะเรียก $\text{Bs}(\text{list}, \text{target}, \text{begin}, \text{mid}-1)$

Bs(list, target, begin, end) =	Bs(list, target, mid+1, end)	; target > list[mid]
	Bs(list, target, begin, mid-1)	; target < list[mid]
	mid	; target == list[mid]
	-1	; begin > end