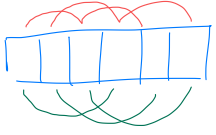


ID _____ Name _____ Sec _____

1 (8 คะแนน) จงตอบคำถามต่อไปนี้สั้น ๆ ไม่ต้องอธิบาย

- $\frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \dots + \frac{n}{n}$ $1 + \frac{1}{2} + \dots + \frac{1}{n} \approx \log n$ = $\Theta(\log n)$
- $1^2 + 2^2 + 3^2 \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$ = $\Theta(n^3)$
- $3^3 + 3^4 + 3^5 \dots + 3^n$ = $\Theta(3^n)$
- $(\frac{1}{5})^1 + (\frac{1}{5})^2 + (\frac{1}{5})^3 + \dots + (\frac{1}{5})^n$ $\frac{\frac{1}{5}(1 - (\frac{1}{5})^n)}{1 - \frac{1}{5}}$ = $\Theta(\frac{1}{5})$
- การเรียงลำดับข้อมูล n ตัวจากมากไปน้อยด้วย mergesort ใช้เวลา = $\Theta(n \log n)$
- การ partition ใน quicksort ใช้เวลา = $\Theta(n)$
- หากเราใช้ median-of-medians of 5 เป็น pivot ในการทำ partition ใน quicksort จะทำให้ quicksort ในกรณี worst case ใช้เวลา = $\Theta(n)$
- ให้ $T(n) = 5T(0.2n) + \Theta(1)$, $T(0) = \Theta(1)$ จะได้ว่า $T(n)$ = $\Theta(n)$

2 (8 คะแนน) จงวิเคราะห์ว่าแต่ละอัลกอริทึมข้างล่างนี้ใช้เวลาเป็น Θ อะไรของตัวแปร n (ให้ถือว่า operator $*$ $/$ $+$ $-$ และอื่น ๆ ใช้เวลา $\Theta(1)$ ทั้งสิ้น และการหาร $/$ ที่แสดงทั้งหมดเป็นการหารแบบปัดเศษทิ้ง) แสดงวิธีทำด้วยในช่องทางขวา

<pre> Hanoi(n, a, b, c) { if (n == 0) return Hanoi(n-1, a, c, b) print(n, a, '-->', c) Hanoi(n-1, b, a, c) } </pre>	$\Theta(2^n)$
<pre> pmod(a, n, m) { p = 1 ak = a while (n > 0) { if (n%2 == 1) p = (p * ak) % m ak = (ak * ak) % m n = n/2 } return p } </pre>	$\Theta(\log n)$
<pre> F(n, p) { if (n < 5) { print(p) } else { for (i=0; i<n; i++) p += i^2 - i p += 2*F(n/2, p) - F(n/2, p/5) } return p/8 } </pre>	$\Theta(n \log n)$
<pre> su(d[1..n], m) { // 1 ≤ m ≤ n s = 0 for (k = 1; k ≤ m; k++) for (i = k; i ≤ n; i += m) s += d[i] return s } </pre> 	$\Theta(n)$

3 (9 คะแนน) จงตอบคำถามต่อไปนี้ในช่องว่างที่กำหนดให้ ไม่ต้องแสดงวิธีทำ 3

- 34 เป็น median-of-median of 5 ของ จำนวนเต็ม 100 ตัว 0,1,2,3,4,..., 99 ได้หรือไม่
ALGO
- Longest common subsequence ของ "XAYYLGZO" กับ "AXQLXGO" คือ
1 1 1 1 1 1 1
- ผลรวมของ maximum contiguous sum ของ [3, -2, 1, -4, 2, -2, 2, -3, 1] คือ
1 2 2 0 2
- การหาจำนวนฟีโบนัชชี F(3) แบบ recursive top-down มีการเรียก F 5 ครั้ง ถ้าหา F(6) จะมีการเรียก F เป็นจำนวน ครั้ง
11
- การ merge ข้อมูล [1, 7, 8, 9] กับ [2,3,4,5] จะเกิดการเปรียบเทียบข้อมูลจาก 2-3 ครั้ง
1 2 3 4 5
- เมทริกซ์ A_1, A_2, A_3 มีขนาด [100×10], [10×100], [100×2] ตามลำดับ ต้องคูณอย่างไรถึงเร็วที่สุด
A(AA)
- start[] = {1,3,0,5,8,5} และ finish[] = {2,4,6,7,9,9} เป็นเวลาเริ่มและจบของการใช้งานห้องห้องหนึ่ง ของงาน 6 งาน จะต้องเลือกงานใดมาใช้ห้องนี้โดยใช้เวลาไม่ซ้อนเหลื่อมกันและได้จำนวนงานมากที่สุด
4
- w[] = {10, 20, 30} และ v[] = {60, 100, 120} เป็นน้ำหนักและมูลค่าของของ 3 ชิ้น ให้มีถุงที่รับน้ำหนักได้มากที่สุด 50 อยากทราบว่าต้องเลือกของชิ้นได้หนักเท่าไร บรรจุลงถุงตาม fractional knapsack problem
60 + 100 + 120 = 280
3
- เอกสารหนึ่งมีตัวอักษร a, b, c, d, e และ f เป็นจำนวน 5, 9, 12, 13, 16 และ 45 ตามลำดับ อยากทราบว่าเราสามารถเข้ารหัสข้อมูลแบบ prefix-free variable length กับข้อมูลทั้งหมดในเอกสารนี้แล้วใช้รวมกับบิตน้อยสุด
72

4 (10 คะแนน) เด็กชายหนึ่ง เล่นเกมสกับเด็กชายสอง โดยมีกติกาดังนี้ เริ่มต้นจะมีรายการของตัวเลขจำนวนเต็มที่ไม่ซ้ำกันอยู่ N ตัว (โดยที่ N เป็นเลขคู่) เด็กสองคนนี้จะต้องผลัดกันเลือกเลขจากรายการนี้ โดยห้ามเลือกเลขที่ถูกเลือกไปแล้ว โดยที่ เด็กชายหนึ่งจะได้เริ่มก่อน โดยผู้ชนะคือผู้ที่ผลรวมของตัวเลขทั้งหมดที่คนนั้นเลือกมีค่ามากกว่า จงอธิบายวิธีการเลือกตัวเลขของเด็กชายหนึ่งที่จะรับประกันว่าเขาจะชนะแน่นอน และ พิสูจน์ด้วยว่าชนะแน่ๆ select the largest value

Greedy optimal sub-structure.

Let S is optimal solution, and G is greedy solution.

Show we can change S to G

$$S = \{a_1, a_2, a_3, \dots, a_k\} \quad ; k \geq 2$$

$$G = \{b_1, b_2, \dots, b_k\}$$

Case 1 $a_k = b_k$ then we can change $\{a_1, \dots, a_{k-1}\}$ to $\{a_1, \dots, b_{k-1}\}$

Case 2 $a_k < b_k$ then

- 5 (10 คะแนน) มีร้านสะดวกซื้อแห่งหนึ่ง จัดรายการส่งเสริมการขายโดยแจกแสตมป์เมื่อซื้อสินค้าบางอย่าง กำหนดให้มีสินค้าทั้งหมด n อย่าง กำกับด้วยหมายเลข 1 ถึง n สินค้าหมายเลข i มีราคา $v[i]$ และจะได้แสตมป์ $s[i]$ ดวงต่อการซื้อสินค้าหมายเลข i 1 ชิ้น เราสามารถซื้อสินค้าแต่ละอย่างเป็นจำนวนเท่าไรก็ได้เท่าที่มีเงินอยู่ นอกจากนี้ ยังมีรายการโบนัสเพิ่มเติมอีกคือ เมื่อซื้อสินค้าทั้งหมดรวมราคาได้มากกว่าหรือเท่ากับ $t[j]$ จะได้แสตมป์เพิ่มอีก $b[j]$ ดวง โดยจะได้เพียงครั้งเดียวที่มียอดค่าใช้จ่ายสูงสุดเท่านั้น ตัวอย่างเช่น ซื้อครบ 100 บาท ได้ 3 ดวง ครบ 150 บาทได้ 5 ดวง กล่าวคือ จากตัวอย่างข้างต้น ถ้าซื้อของรวมเป็นราคา 255 บาท จะได้แสตมป์ 5 ดวงเท่านั้น (ไม่ใช่ 8 ดวง) กำหนดให้มีรายการโบนัสทั้งหมด m รายการ เรามีเงินอยู่ในมือทั้งหมด P บาท จงหาวิธีใช้เงิน P บาทเพื่อให้ได้แสตมป์มากที่สุด (กำหนดให้ $v[i]$ และ $t[i]$ เป็นจำนวนเต็มบวก และ $s[j]$, $b[j]$ เป็นจำนวนเต็มไม่ลบทั้งหมด)

- จงเขียน recurrent relation เพื่อคำนวณจำนวนแสตมป์สูงสุดที่เป็นไปได้ พร้อมทั้งระบุเงื่อนไขเริ่มต้นทั้งหมดด้วย
- จงอธิบายว่าถ้านำ recurrent relation นั้นไปเขียนเป็นโปรแกรม จะมีประสิทธิภาพในการทำงานเป็นเท่าไร พร้อมอธิบายเหตุผล

$O(nP)$?

$MaxStamp (curMoney) = get Bonus Stamp (P - curMoney)$

array value .
↓
if min (v) > curMoney

= max ($s[i] + MaxStamp (curMoney - v[i])$ for i such that $v[i] < curMoney$) ; else.

	V	1	10	15	20						
S		2	3	3	7						
0	1	2	3	4	5	6	7	8	9	10	11
0	2	4	6	8	10	12	14	16	18	20	22