01204512: การบ้าน 1

กำหนดส่ง 8 ส.ค. 2555

1. Prove that
$$\frac{n!}{n^n} \leq \frac{1}{2^{\lfloor n/2 \rfloor}},$$
for any integer $n \geq 1$.

   (Note that $\lfloor x \rfloor$ is the largest integer no greater than x, e.g., $\lfloor 10 \rfloor = 10$ and $\lfloor 1.2 \rfloor = 1$.)

2. Recall that a *prime number* is a positive integer with exactly two positive factors which is 1 and itself. Prove that for any integers a and b, if a prime number p divides ab, then p must divide a or b (or both).

   (Note that this is not trun if p is not prime. For example, 4 divides $60 = 6 \cdot 10$, but it does not divide either 6 or 10.)

3. Consider the following programs and analyze their running times.

   (a)
   ```
   r ← 0
   for i = 1, 2, ... n
       for j = 1, 2, ..., n
           r ← i + j
   ```
   Analyze the running time in terms of n.

   (b)
   ```
   r ← 0
   for i = 1, 2, ... n
       for j = 1, 2, ..., m
           r ← i + j
   ```
   Analyze the running time in terms of n and m.

   (c)
   ```
   r ← 0
   for i = 1, 2, ... n
       k ← 1
       while k^2 ≤ m
           r ← i + j
           k ← k + 1
   ```
   Analyze the running time in terms of n and m.

   (d)
   ```
   r ← 0
   k ← 1
   for i = 1, 2, ... n
       while k^2 ≤ n
           r ← i + j
           k ← k + 1
   ```
   Analyze the running time in terms of n.

4. Consider the following sorting algorithm for sorting n items. For simplicity assume that n is a square, i.e., there exists a positive integer k such that $k^2 = n$.

   Given n items in an array, the algorithm proceeds as follows.

   (a) The algorithm divides all items into $k = \sqrt{n}$ groups; each group contains n/k items.

   (b) For each group, the algorithm uses insertion sort to sort items in it. The sorted items for each group are stored in a list.

   (c) It then employs a k-way merge. It looks at the head of each sorted list, picks the smallest, and moves it to the solution list.

   Analyze the running time of steps (a), (b), and (c), in terms of n and conclude with the total running time of the algorithm.

5. จงแก้สมการ recurrence ต่อไปนี้ ให้อยู่ในรูปสัญกรณ์ Big-O (สมมติให้ $T(1) = 1$)
   (a) $T(n) = n + T(n-1)$
   (b) $T(n) = n^2 + T(n-1)$
   (c) $T(n) = n + T(3n/4)$
   (d) $T(n) = 2n + 2T(n/2)$
   (e) $T(n) = n^2 + 4T(n/3)$

6. **Closest pairs.** เราจะออกแบบอัลกอริทึมที่รับเซตของจุด n จุดบนระนาบ จากนั้นหาคู่ของจุดที่อยู่ใกล้กันมากที่สุด

   เราจะเรียกจุดดังกล่าวเป็น $p_1, p_2, \ldots, p_n$ โดยสำหรับ $1 \leq i \leq n$, จุด p_i มีพิกัดเป็น (x_i, y_i) ระยะระหว่างจุด p_i และจุด p_j เขียนแทนด้วย $d(p_i, p_j)$ มีค่าเท่ากับ $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ นอกจากนี้, ให้เซต $P = \{p_1, \ldots, p_n\}$ แทนเซตของจุดทั้งหมด

   นอกจากนี้เราสมมติว่าเซตของจุดดังกล่าว เรียงตามพิกัดในแนวแกน x นั่นคือ $x_i \leq x_{i+1}$ สำหรับทุก ๆ $1 \leq i < n$ และเรายังมีลิสต์ Q ของดัชนีของจุดที่เรียงตามลำดับพิกัดในแนวแกน y ด้วย

   เพื่อความง่ายในการวิเคราะห์เราจะสมมติว่า n เป็นกำลังของ 2

   (a) จงอธิบายอัลกอริทึมที่ทำงานในเวลา $O(n^2)$ ที่หาคู่ของจุด p_i และ p_j ที่อยู่ใกล้กันที่สุด

   เราจะออกแบบอัลกอริทึมแบบ divide and conquer ดังนี้ เราจะแบ่งจุดออกเป็นสองกลุ่มตามแนวแกน x นั่นคือ เราจะมีเซตของจุด $L = \{p_1, p_2, \ldots, p_{n/2}\}$ และ จุด $R = \{p_{n/2+1}, p_{n/2+2}, \ldots, p_n\}$ จากนั้นเราจะเรียกตัวเองเพื่อหาคู่ของจุดที่ใกล้กันมากที่สุดในเซต L และ R เรียกคู่ของจุดที่ใกล้กันที่สุดในเซต L ว่าเป็น p_{l1} และ p_{l2} ให้ทั้งคู่ห่างกัน Δ_L หน่วย, ในทำนองเดียวกัน เรียกคู่ของจุดที่ใกล้กันที่สุดในเซต R ว่าเป็น p_{r1} และ p_{r2} ให้ทั้งคู่ห่างกัน Δ_R หน่วย ให้ $\Delta = \min\{\Delta_L, \Delta_R\}$ แทนระยะสั้นที่สุดที่เราได้จากการเรียกตัวเองไปแก้ปัญหาย่อย

   (b) มีกรณีใดบ้างที่ระยะห่างที่ใกล้ที่สุดของจุดสองจุดใน P จะมีค่าน้อยกว่า Δ

   ในส่วนถัดไปเราจะพยายามหาคู่ของจุดที่สอดคล้องกับกรณีที่เราถามในข้อ (b)

   (c) สมมติว่าเวลาที่เราใช้ในการแก้ปัญหาดังกล่าวเมื่อจุดเริ่มต้นมี n จุด คือ $R(n)$ นอกจากนี้ให้ $T(n)$ แทนเวลาที่อัลกอริทึมนี้ใช้ในการแก้ปัญหา closest pair จงเขียน recurrence ของ $T(n)$ ถ้าเราต้องการให้ $T(n) = O(n \log n)$ ฟังก์ชัน $R(n)$ จะต้องเป็นอย่างไร?

   เราต้องการหาคู่ของจุด $p_l \in L$ และ $p_r \in R$ ที่ (1) $d(p_l, p_r) < \Delta$ และ (2) ระยะห่างระหว่าง p_l และ p_r น้อยที่สุด

   ในส่วนถัดไปเราจะแสดงวิธีการหาคู่ของจุด p_l และ p_r ดังกล่าว (ถ้ามี)

   (d) ถ้าเราใช้การทดลองหาทุกคู่ที่เป็นไปได้ เวลาที่ใช้เป็นเท่าใด?

   สังเกตว่าในการแบ่งข้อมูลเริ่มต้น P ออกเป็นสองกลุ่ม L และ R เราแบ่งโดยใช้พิกัดบนแกน x กล่าวคือ เราจะมีค่า M และเส้นตรง D ที่ระบุด้วยสมการ $x = M$ ที่ขนานกับแกน y ที่แบ่งกลุ่มข้อมูลได้เป็นสองกลุ่ม กล่าวคือ จุดในเซต L จะมีพิกัดบนแกน x น้อยกว่าหรือเท่ากับ M และทุกจุดในเซต R จะมีพิกัดบนแกน x มากกว่าหรือเท่ากับ M

   (e) ให้พิสูจน์ว่า:
      (1) ทุก ๆ จุด $p_i \in L$ ที่ $x_i \leq M - \Delta$ ไม่สามารถเป็นคำตอบ p_l ที่เราต้องการหาได้, และ
      (2) ทุก ๆ จุด $p_i \in R$ ที่ $x_i \geq M + \Delta$ ไม่สามารถเป็นคำตอบ p_r ที่เราต้องการหาได้

ข้อ (e) รับประกันว่าเราจำเป็นจะต้องพิจารณาจุดที่อยู่ในระยะห่าง Δ จากเส้นตรง D เท่านั้น อย่างไรก็ตาม เป็นไปได้ที่จุดที่อยู่ในของเขตดังกล่าวอาจจะมีเป็นจำนวนมาก

(f) พิจารณาจุด $p_i \in L$ ที่ $x_i \geq M - \Delta$ จงแสดงว่าจุด $p_j \in R$ ที่จะสามารถจับคู่เป็นคำตอบกับ p_i ได้ จะต้องมีพิกัด y_j สอดคล้องกับ $y_i - \Delta \leq y_j \leq y_i + \Delta$ เท่านั้น

ต่อไปเราจะพิสูจน์ว่าสำหรับจุด $p_i \in L$ ที่ $x_i \geq M - \Delta$ จะมีจุด $p_j \in R$ ที่สามารถจับคู่เป็นคำตอบกับ p_i ได้ไม่เกิน 8 จุด

(g) พิจารณาสี่เหลี่ยมจัตุรัส A ขนาดกว้าง Δ หน่วย ให้เซต S เป็นเซตของจุดที่อยู่ใน A ที่ทุก ๆ คู่ของจุดใน S มีระยะห่างกันไม่น้อยกว่า Δ หน่วย จงพิสูจน์ว่า เซต S จะมีสมาชิกได้ไม่เกิน 4 จุด

(hint: พิจารณากล่องขนาด $\Delta/2 \times \Delta/2$ ใช้ Pigeonhole principle)

(h) ใช้ความจริงที่พิสูจน์จากข้อ (f) และ (g) เพื่อแสดงว่าสำหรับจุด $p_i \in L$ ที่ $x_i \geq M - \Delta$ จะมีจุด $p_j \in R$ ที่สามารถจับคู่เป็นคำตอบกับ p_i ได้ไม่เกิน 8 จุด จากนั้นให้แสดงว่าจำนวนคู่ของจุดทั้งหมดที่จำเป็นต้องพิจารณาเพื่อจะหา p_l และ p_r มี $O(n)$

จากข้อ (h) เราจะได้ว่า แทนที่เราจะต้องพิจารณาคู่ของจุดใน L และ R ทุกคู่ (ซึ่งมีจำนวน $\Omega(n^2)$) เราจำเป็นต้องพิจารณาเพียงแค่ $O(n)$ จุดเท่านั้น ดังนั้น ถ้าเราดำเนินการดังกล่าวได้รวดเร็ว เราก็สามารถหาอัลกอริทึมของปัญหา closest pair ที่ทำงานในเวลารวมได้เป็น $O(n \log n)$

(i) จงอธิบายอัลกอริทึมที่รับ L, R, Δ รวมทั้ง Q ที่หาคู่ของจุด p_l และ p_r ที่ทำงานในเวลา $O(n)$

(hint: ไล่พิจารณาจากบนลงล่าง)

(j) (โบนัส) สังเกตว่าอัลกอริทึมที่เราอธิบายมา ยังไม่สามารถทำงานได้จริงเนื่องจากอัลกอริทึมต้องการลิสต์ Q ซึ่งเรายังไม่ได้ระบุว่า เมื่อเรียกตัวเองเพื่อแก้ปัญหาในเซต L และ R จะสร้างลิสต์ดังกล่าวได้อย่างไร ในข้อนี้ให้อธิบายการวิธีการสร้างรายการ Q สำหรับปัญหาย่อยทั้งสอง