การบ้านวิชา Algorithm Design เทอมต้น ปีการศึกษา 2555 ครั้งที่ 1

กำหนดส่ง:

ให้ post link ไปยัง youtube ไว้ที่ https://www.facebook.com/groups/274760319210579/ ภายในวันที่ 2 ก.ค. 2555 (นับเวลาที่ post ใน facebook group)

กฎ:

1. แต่ละกลุ่มให้ทำเพียงหนึ่งข้อ
2. ห้ามทำปัญหาข้อเดียวกันซ้ำกัน ยกเว้นคำตอบที่ได้มีความแตกต่างอย่างมีนัยยะสำคัญ
    a. อย่างไรก็ตาม ปัญหาหนึ่งข้อห้ามทำเกินซ้ำกันเกิน 3 กลุ่ม (ถึงแม้จะมีความแตกต่างกัน ก็ห้ามมีกลุ่มที่ 4, 5,...)
3. ขอให้เน้นที่ทำให้เพื่อน ๆ ฟังเฉลยของคุณแล้วรู้เรื่อง

ปัญหา:

1. กำหนดให้ปัญหาการหาผลรวมย่อยของ เมทริกซ์ เป็นดังนี้ กำหนดให้มีเมทริกซ์ A ขนาด m แถว n คอลัมน์ และ a_{ij} คือสมาชิกในแถวที่ i คอลัมน์ที่ j ของเมทริกซ์ดังกล่าว ปัญหาการหาผลรวมย่อยคือการหาค่า F(x1,y1,x2,y2) โดยกำหนดให้ F(x1,y1,x2,y2) มีค่าเท่ากับ $\sum_{i=y1}^{y2}\sum_{j=x1}^{x2} a_{ij}$ จงออกแบบขั้นตอนวิธีสำหรับการหาค่า F(x1,y1,x2,y2) พร้อมทั้งวิเคราะห์ประสิทธิภาพในการทำงาน
2. Unlike a decreasing geometric series, the sum of the harmonic series $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$ diverge; that is $\sum_{i=1}^{\infty}\frac{1}{i} = \infty$. It turns out that, for large n, the sum of the first n temrs of this series can be well approximated as $\sum_{i=1}^{n}\frac{1}{i} \approx \ln(n) + \gamma$ where ln is natural logarithm and γ is a particular constant 0.57721.... Show that $\sum_{i=1}^{n}\frac{1}{i} = \Theta(\log n)$ (Hint: To show an upper bound, decrease each denominator to the next power of two. For a lower bound, increase each denominator to the next power of 2.) (this is problem 1.5 in [1])
3. Consider this algorithm that multiplies two integers

```
function multiply(x, y)
Input: Two integers x and y, where y >= 0
Output: Their product

if y = 0 return 0
z = multiply(x, ⌊y/2⌋)
if y is even
    return 2z
else
    return x + 2z
```

How long does the recursive multiplication algorithm take to multiply an n-bit number by an m-bit number (this is problem 1.7 in [1])

4. positive integer N is a power if it is of the form q^k, where q, k are positive integers and k > 1.
   (a) Give an ef cient algorithm that takes as input a number N and determines whether it is a square, that is, whether it can be written as q2 for some positive integer q. What is the running time of your algorithm?
   (b) Show that if N = q^k (with N, q, and k all positive integers), then either k <= log(N) or N = 1.
   (c) Give an efficient algorithm for determining whether a positive integer N is a power. Analyze its running time. (this is problem 1.32 in [1])
5. http://thailandoi.org/files/TOI_C_01-2009_(01_star).pdf
6. http://thailandoi.org/files/TOI_CPP_03-2009_02_riddle.pdf
7. http://thailandoi.org/files/TOI_C_04-2009_01_diary.pdf
8. http://thailandoi.org/files/TOI_C_04-2009_02_house.pdf
9. http://thailandoi.org/files/TOI_CPP_05-2009_01_cocktail.pdf
10. http://thailandoi.org/files/ytopc-dec08-wheel.pdf
11. http://thailandoi.org/files/ytopc-dec08-hands.pdf
12. http://thailandoi.org/files/ytopc-feb09-atom.pdf
13. จงออกแบบอัลกอริทึมสำหรับการหาค่าฐานนิยมของ array A ซึ่งมีขนาด n ช่อง พร้อมทั้งวิเคราะห์ประสิทธิภาพในการทำงาน
14. The Sieve of Erastothenes is a method used to compute all primes less than n. We begin by making a table of integers 2 to n. We find the smallest integer I, that is not crossed out, print I and cross out i, 2i, 3i, ... When i > $\sqrt{n}$. The algorithm terminates. What is the running time of this algorithm? (This is problem 2.14 in [2])
15. Find two functions f(n) and g(n), both monotonically increasing, such that f(n) $\notin$ O(g(n)) and g(n) $\notin$ O(f(n)) (This is problem 3.17 in [3])

Use this story for the next two problems:

You're doing some stress-testing on various models of glass jars to determine the heights from which they can be dropped and still not break. The setup for this experiment, on a particular type of jar, is as follows. You have a ladder with n rungs, and you want to find the highest rung from which you can drop a copy of the jar and not have it break. We call this the highest safe rung.

It might be natural to try binary search: drop a jar from the middle rung, see if it breaks, and then recursively try from rung n/4 or 3n/4 depending on the outcome. But this has the drawback that you could break a lot of jars in finding the answer.

If your primary goal were to conserve jars, on the other hand, you could try the following strategy. Start by dropping a jar from the first rung, then the second rung, and so forth, climbing on higher each time until the jar breaks. In this way, you only need a single jar – at the moment it breaks, you have to correct answer – but you may have to drop it n times (rather than log n as in the binary search solution).

So here is the trade-off: it seems you can perform fewer drops if you're willing to break more jars. To understand better how this trade-off works at a quantitative level, let's consider how to run this experiment given a fixed "budget" of k >=1 jars. In other words, you have to determine the correct answer – the highest safe rung – and can use at most k jars in doing so.

16. Suppose you have a budget of k = 2 jars. Describe a strategy for finding the highest safe run that requires you to drop a jar at most f(n) times, for some function f(n) that grows slower than linearly. (In other words, it should be the case that $\lim_{n\to\infty} f(n)/n = 0$ (This is problem 8(a) in chapter 2 of [4])
17. Now suppose you have a budget of k > 2 jars, for some given k. Describe a strategy for finding the highest safe run using at most k jars. If $f_k(n)$ denotes the number of times you need to drop a jar according to your strategy, then the function $f_1, f_2, f_3, \ldots$ should have the property that each grows asymptotically slower than the previous one: $\lim_{n\to\infty} f_k(n)/f_{k-1}(n) = 0$ (This is problem 8(b) in chapter 2 of [4])

Reference

[1] Algorithms, S. Dasgupta, C. Papadimitriou, and U.V. Vazirani, McGraw-Hill, 2007

[2] Data Structures and Algorithm Analysis, Mark A Weiss, Addison Wesley, 1992

[3] Introduction to Algorithms A Creative Approach, Udi Manber, Addison Wesley, 1989

[4] Algorithm Design, Jon Kleinberg, Éva Tardos, Addison Wesley, 2005