MA/CSSE 473 – Design and Analysis of Algorithms **In-class Quiz Day 07**

Name:_____**Solution**_____ Score:____/ 9 circle your Section # 01(3rd) 02 (4th)

1. Prove that there is always a survivor in an odd pie fight.

Base case. P(1), 3 people. Suppose A and B are the closest pair, and C is the third person. Since all of the distances are different, the distances between A and C, and between B and C are strictly larger than the distance between A and B. Thus A and B throw pies at each other, and C is the survivor.
Induction step. Assume that P(k) is true (in every pie fight with 2k+1 people, there is a survivor). Show that P(k+1) is true (in every pie fight with 2K+3 people, there is a survivor).
Let A and B be the closest pair in this group of 2k+3 people. They throw pies at each other.
If someone else throws a pie at one of them, then for the remaining 2k+1 people, there are only 2k pies to hit them, so someone survives.
If no one else throws a pie at A or B, then the other people comprise a 2k+1 pie fight, which has a survivor, by the induction assumption.

2. When exponentiating n-bit numbers x^y (mod N), where N is also n-bit, how many recursive calls are needed? **n**

3. Each call is Θ(**n**)

4. Entire exponentiation algorithm is Θ($\mathbf{n^2}$)

5. Show the recursive calls for Euclid's Algorithm applied to a=188 and b=144.

   gcd(188, 144) = gcd(144, 44) = gcd(44, 12) = gcd(12, 8) = gcd(8, 4) = gcd(4, 0) = 4.

6. The following two conditions imply that d = gcd(a,b):

   a. d divides both a and b

   b. d = ax + by for some integers x and y

7. Prove the validity of the extended Euclid algorithm.

```
def euclidExtended(a, b):
    """ INPUT:  Two integers a and b with a >= b >= 0
        OUTPUT: Integers x, y, d such that d = gcd(a, b)
                and d = ax + by"""
    print ("    ", a, b) # so we can see the process.
    if b == 0:
        return 1, 0, a
    x, y, d =  euclidExtended(b, a % b)
    return y, x - a//b*y, d
```

Solution: First, the number d it produces really is the gc
of a and b. We can just ignore the x and y values,
and we have the same algorithm as before.

- We must show that the x and y it returns are such that ax + by =d.
- We do that by induction on b.
  - **Base case:** b=0
    Then gcd(a,b) = a,
    and the algorithm
    produces x = 1,
    y = 0. ☑
  - **Induction step:** b > 0.
    - It finds gcd(a, b) by calling euclidExtended(b, a%b)
    - Since a%b is smaller than b, by induction the x' and y' returned by the recursive call are such that
      gcd(b, a %b) = bx'+ (a%b)y'
    - We can write a % b as a – ⌊a/b⌋ * b
    - d = gcd(a, b) = gcd(b, a%b) = bx' + (a%b)y'
      = bx' +(a – ⌊a/b⌋ * b)y' = ay' +b(x' - ⌊a/b⌋y')
    - Thus x = y' and y = x' - ⌊a/b⌋y' are the numbers that make ax + by = d

This x and y are the numbers returned by the algorithm

8. What became clear to you as a result of today's discussion? (or write N/A)

Must have an answer; any answer is okay

9. Is there anything from today's discussion that was unclear, do you have questions, or is there anything else you want to tell me? (or write N/A)

Must have an answer; any answer is okay