Algorithms: GCD analysis

Review questions (7 minutes)

> *Notice that section headings will often contain a target time, like the one above. The manager may want to set a timer to help ensure your group is making good progress through the section in order to finish on time.*

1 What is 27 mod 5?

2 What is 2 mod 5?

3 Which of the following statements is always true, assuming that a and b are positive integers?

- $0 \leq a \bmod b < b$
- $0 \leq a \bmod b < a$

4 What is 5 mod 0?

5 Is 0 divisible by 10?

> *When you see a STOP sign like the one below, it means that you should wait until instructed to go on to the next page. If you finish before other groups, take the time to go back over your answers on this section: are there any lingering questions or confusions you have? Any answers you could make more complete or more nuanced? Anything you notice, or wonder about?*

Model 1: GCD (8 minutes)

Definition 1. Recall that the *greatest common divisor*, or GCD, of two positive integers a and b is defined as the largest positive integer which evenly divides both a and b. The GCD of a and b is denoted $\gcd(a, b)$.

6 What is $\gcd(12, 30)$?

7 What are the prime factorizations of 12 and 30?

8 What do the prime factorizations of 12 and 30 have to do with $\gcd(12, 30)$?

9 What is $\gcd(144, 690)$?

10 What if we extend the definition of GCD to apply to all nonnegative integers? What should $\gcd(a, 0)$ be when $a > 0$?

11 Have the Reflector take a minute to share one way that your group has worked well together, and one way that you could improve.

© 2022 Brent A. Yorgey. This work is licensed under a Creative Commons Attribution 4.0 International License.

Model 2: The Euclidean Algorithm

Consider the four algorithms specified below. They are all supposed to compute the GCD of nonnegative integers, but only two of them are correct.

```
GCDIterA(m,n) =
  a ← m
  b ← n
  while (a ≠ 0)
    if a ≤ b
      then b ← b mod a
      else a ← a mod b
  if a = 0 then return b else return a
```

```
GCDIterB(m,n) =
  a ← m
  b ← n
  while (a ≠ 0) and (b ≠ 0)
    if a ≤ b
      then b ← b mod a
      else a ← a mod b
  if a = 0 then return b else return a
```

```
GCDRecA(a,b) =
  if b = 0
    then a
    else GCDRecA(b, a mod b)
```

```
GCDRecB(a,b) =
  if b = 0
    then a
    else GCDRecB(a mod b, b)
```

12 Trace the execution of each algorithm on the inputs $(144, 690)$. *It may be tempting to split these up and have one person do each independently. However, that rarely makes things any faster, and I recommend resisting that temptation. Instead, work together as a group on one at a time, and build a shared understanding of what is going on.*

© 2022 Brent A. Yorgey. This work is licensed under a Creative Commons Attribution 4.0 International License.

13 What do you think Iter and Rec stand for in the names of the functions?

14 List some similarities and differences among the algorithms.

15 Which algorithms are incorrect? What is wrong with them?

16 For the correct algorithms, describe in a few sentences what happened to the values of a and b as the algorithm ran. Can you explain why the algorithms will always stop eventually?

17 Look at one of your execution traces from Question 12. Find the gcd of a and b after each iteration of the algorithm. What do you notice?

© 2022 Brent A. Yorgey. This work is licensed under a Creative Commons Attribution 4.0 International License.