Tiling A Floor

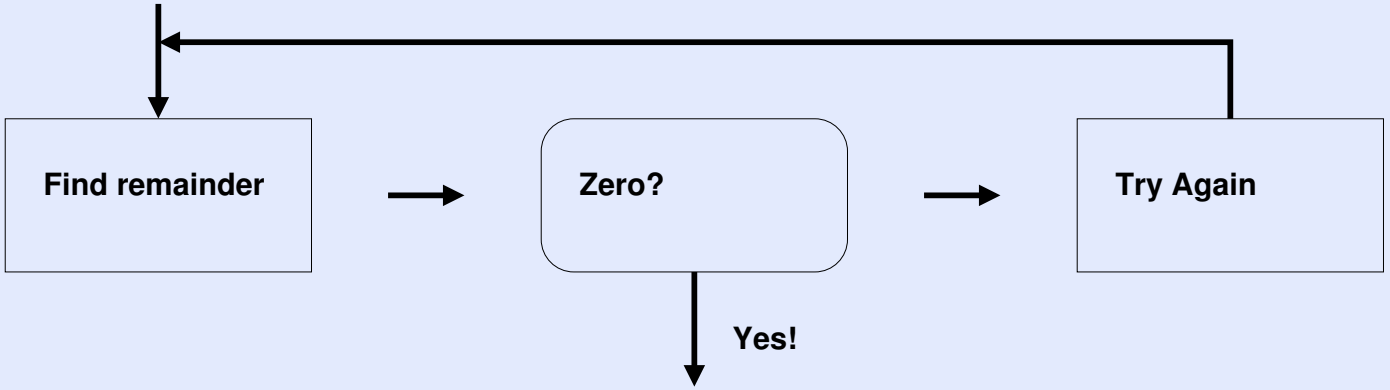
- A room of dimension m by n is given. You are asked to *tile* the room with the constraints
  - Tile must be square
  - Minimum number of tiles must be used
- Although obvious from the problem workding, I emphasize here that you cannot leave any empty space in the floor
- Example: For m = 119, n = 544, the answer is 17
- The solution

Tiling A Floor

- A room of dimension m by n is given. You are asked to *tile* the room with the constraints
  - Tile must be square
  - Minimum number of tiles must be used
- Although obvious from the problem workding, I emphasize here that you cannot leave any empty space in the floor
- Example: For m = 119, n = 544, the answer is 17
- The solution
  - Find a number such that it evenly divides both x and y
  - Find the largest such number
- Compute the greatest common divisor (or highest common factor)

Celebrated Euclid's Algorithm

- Work out the following *algorithm*.
  1. (Find remainder.) Divide m by n and let r be the remainder.
  2. (Even Division?) If $r == 0$, return with n as the answer.
  3. (Try again.) Set $m = n$, $n = r$ and go to step 1.
- For m = 119, n = 544, r is the sequence 68, 51, 17, 0.
- Concept of flowchart



Algorithm Characteristics

- Finite. An algorithm must always terminate.
- Definite. Each step is precisely defined. This is the principal reason for us to have computer languages.
- Primitives. Each step must be “reasonably” simple.
- Input. An algorithm has zero or more inputs.
- Output. An algorithm has one or more outputs. It is useful to *prove* that the output is what is intended.

Algorithm Characteristics

- Finite. An algorithm must always terminate.
- Definite. Each step is precisely defined. This is the principal reason for us to have computer languages.
- Primitives. Each step must be “reasonably” simple.
- Input. An algorithm has zero or more inputs.
- Output. An algorithm has one or more outputs. It is useful to *prove* that the output is what is intended.
  If r equals zero, then we are done. Otherwise, since $m = qn + r$ for some integer q, any number that divides m and n will also divide r.

Algorithm Characteristics

- Finite. An algorithm must always terminate.
- Definite. Each step is precisely defined. This is the principal reason for us to have computer languages.
- Primitives. Each step must be “reasonably” simple.
- Input. An algorithm has zero or more inputs.
- Output. An algorithm has one or more outputs. It is useful to *prove* that the output is what is intended.
  If r equals zero, then we are done. Otherwise, since $m = qn + r$ for some integer q, any number that divides m and n will also divide r. Further, any number that divides n and r will also divide m.

Algorithm Characteristics

- Finite. An algorithm must always terminate.
- Definite. Each step is precisely defined. This is the principal reason for us to have computer languages.
- Primitives. Each step must be “reasonably” simple.
- Input. An algorithm has zero or more inputs.
- Output. An algorithm has one or more outputs. It is useful to *prove* that the output is what is intended.
  If r equals zero, then we are done. Otherwise, since $m = qn + r$ for some integer q, any number that divides m and n will also divide r. Further, any number that divides n and r will also divide m. So the set of numbers that divide m and n are precisely the set of numbers that divide n and r. QED.