CSCI-UA 480.4: APS
Algorithmic Problem Solving

Bitmasks

Instructor: Joanna Klukowska

created based on materials for this class by
Bowen Yu and materials shared by the authors of
the textbook Steven and Felix Halim

Questions

- homework 2 questions?
- any other questions?

Bits, bit operations, bit masks

Bit operations

Bit operations

left shift <<

Bit operations

left shift <<

right shift >>

Bit operations

left shift <<

right shift >>

- shifting is equivalent to multiplying by powers of two (but faster)
- WARNING: avoid shifting out of the type range

Bit-mask

A **bit mask** of the form has one bit on (i.e., equal to 1) in the th position and all other bits off (i.e., equal to 0).

- to determine if a bit at the th position in a particular value is on or off, we can this bit mask and the bitwise and operator:

  example:

  What does this code do?

Bit-mask

A **bit mask** of the form has one bit on (i.e., equal to 1) in the th position and all other bits off (i.e., equal to 0).

- to determine if a bit at the th position in a particular value is on or off, we can this bit mask and the bitwise and operator:

  example:

  What does this code do?

  It prints the binary represetnation of .

Bit-mask continued

Modifying bits within a value:

- set the th bit to be on (regardless of what it was before): ???
- set the th bit to be off (regardless of what it was before): ???
- set the th bit to be on if it is currently off, and to be off if it is currently on (just flip that bit to the opposite): ???

Bit-mask continued

Modifying bits within a value:

- set the th bit to be on (regardless of what it was before): ???
- set the th bit to be off (regardless of what it was before): ???
- set the th bit to be on if it is currently off, and to be off if it is currently on (just flip that bit to the opposite): ???
- invert all bits after the last one bit: ???
- set the last one bit in x to 0: ???
- test if x is a power of two: ???

Bit-mask continued

Modifying bits within a value:

- set the th bit to be on (regardless of what it was before):
- set the th bit to be off (regardless of what it was before):
- set the th bit to be on if it is currently off, and to be off if it is currently on (just flip that bit to the opposite):
- invert all bits after the last one bit:
- set the last one bit in x to 0:
- test if x is a power of two:

Bit-mask continued

Modifying bits within a value:

- set the th bit to be on (regardless of what it was before):
- set the th bit to be off (regardless of what it was before):
- set the th bit to be on if it is currently off, and to be off if it is currently on (just flip that bit to the opposite):
- invert all bits after the last one bit:
- set the last one bit in x to 0:
- test if x is a power of two:

is a 32-bit mast (1 is an int)

to use a 64-bit bit mast: use (1**L** << k)