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

Useful library functions, C/C++

- : the number of zeros at the beginning of the bit representation
- : the number of zeros at the end of the bit representation
- : the number of ones in the bit representation
- : the parity (even or odd) of the number of ones in the bit representation

Useful library functions, C/C++

- : the number of zeros at the beginning of the bit representation
- : the number of zeros at the end of the bit representation
- : the number of ones in the bit representation
- : the parity (even or odd) of the number of ones in the bit representation

example

Useful library functions, C/C++

- : the number of zeros at the beginning of the bit representation
- : the number of zeros at the end of the bit representation
- : the number of ones in the bit representation
- : the parity (even or odd) of the number of ones in the bit representation

example

Add suffix to the above functions to use the versions instead of .

Useful library functions, Java

in class :

- Returns the number of one-bits in the two's complement binary representation of the specified int value.
- Returns an int value with at most a single one-bit, in the position of the highest-order ("leftmost") one-bit in the specified int value.
- Returns an int value with at most a single one-bit, in the position of the lowest-order ("rightmost") one-bit in the specified int value.
- Returns the number of zero bits preceding the highest-order ("leftmost") one-bit in the two's complement binary representation of the specified int value.

...

Exercises

- Write a code fragment that sets (turns on) every bit in an even position in an integer mask. (Can you define such a mask in one *step*?)
- Compute the *distance* between two bitmasks, i.e., how many positions in the two values are different? (this is known as the hamming distance)

Exercises

- Write a code fragment that sets (turns on) every bit in an even position in an integer mask.

- Compute the *distance* between two bitmasks, i.e., how many positions in the two values are different? (this is known as hamming distance)

Sets

Representing sets

Any subset of a set

can be represented by an bit integer. The bits of a number indicate if the element is present in the set or not.

Example

represents the subset {0, 1, 3, 8, 11}

To create such a subset representation use code as follows:

and to print the size of such a subset, use

Set operations

How can the following operations be performed on the sets represented by integers:

- set intersection (what do the two sets have in common)
- set union (all elements in either one, or both sets)
- set difference (all elements in A that are not in B)

Set operations

How can the following operations be performed on the sets represented by integers:

- set intersection (what do the two sets have in common)
- set union (all elements in either one, or both sets)
- set difference (all elements in A that are not in B)

Assume that two integers and represent two sets (max number of elements in each set is 32).

- is the intersection
- is the union
- is the difference

Exercises

- Create two sets and . Compute their union, intersection and difference. For each print the content of the set and its size.
- List all possible subsets of a set .
- List all possible subsets of a set that have exactly 7 elements.

Library Classes

Library Classes, C++

_____ in C++

Bit access

- Access bit
- Count bits set
- Return size
- Return bit value
- Test if any bit is set
- Test if no bit is set

Bit operations

- Set bits
- Reset bits
- Flip bits
- overloaded operators (see code example from Cplusplus.com on the side)
- ...

Library Classes, Java

in Java

- Sets all of the bits in this BitSet to false.
- Sets the bit at the specified index to the complement of its current value.
- Returns the value of the bit with the specified index.
- Returns true if the specified BitSet has any bits set to true that are also set to true in this BitSet.
- Returns the index of the first bit that is set to true that occurs on or after the specified starting index.
- Sets the bits from the specified fromIndex (inclusive) to the specified toIndex (exclusive) to the specified value.
- ...

Example Application:
Prime Numbers

Prime numbers

Task: Generate all prime numbers in the range from 0 to N

Prime numbers

Task: Generate all prime numbers in the range from 0 to N

Very naive algorithm:

Less naive algorithm:

Prime numbers

Task: Generate all prime numbers in the range from 0 to N

Very naive algorithm:

Less naive algorithm: -

Sieve of Eratosthenes

-

- Algorithm:
  - set all values in the range to *probably prime*
  - set 0 and 1 to be ~~not prime~~
  - for p in 2:N
    - if p is *probably prime*
      - set p to **definitely prime**
      - set all multiples of p (except for p itself) to ~~not prime~~

Sieve of Eratosthenes

- Example: N = 20
  - ~~0~~ ~~1~~ *2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20*
  - ~~0~~ ~~1~~ **2** *3* ~~4~~ *5* ~~6~~ *7* ~~8~~ *9* ~~10~~ *11* ~~12~~ *13* ~~14~~ *15* ~~16~~ *17* ~~18~~ *19* ~~20~~
  - ~~0~~ ~~1~~ **2** **3** ~~4~~ *5* ~~6~~ *7* ~~8~~ ~~9~~ ~~10~~ *11* ~~12~~ *13* ~~14~~ ~~15~~ ~~16~~ *17* ~~18~~ *19* ~~20~~
  - ~~0~~ ~~1~~ **2** **3** ~~4~~ **5** ~~6~~ *7* ~~8~~ ~~9~~ ~~10~~ *11* ~~12~~ *13* ~~14~~ ~~15~~ ~~16~~ *17* ~~18~~ *19* ~~20~~
  - ~~0~~ ~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ *11* ~~12~~ *13* ~~14~~ ~~15~~ ~~16~~ *17* ~~18~~ *19* ~~20~~
  - ~~0~~ ~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ *13* ~~14~~ ~~15~~ ~~16~~ *17* ~~18~~ *19* ~~20~~
  - past the halfway point, all the remaining probably primes are definitely prime
    ~~0~~ ~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ **13** ~~14~~ ~~15~~ ~~16~~ **17** ~~18~~ **19** ~~20~~

Sieve of Sundaram (1934)

- Finds all the primes in the list of integers from 1 to 2N+2.
- Algirthm:
    - add to the set of primes
    - mark all values of the form:

      (where , ,) as ~~not prime~~
    - for each remaining unmark value, double it and add 1, and add it to the set of primes (these are all odd primes between 3 and 2N+2)

Sieve of Sundaram (1934)

- Finds all the primes in the list of integers from 1 to 2N+2.
- Algirthm:
  - add to the set of primes
  - mark all values of the form:

    (where , ,) as ~~not prime~~
  - for each remaining unmark value, double it and add 1, and add it to the set of primes (these are all odd primes between 3 and 2N+2)

- Example: N = 10
  - 1 2 3 4 5 6 7 8 9 10
  - i = 1, j = 1
1 2 3 ~~4~~ 5 6 7 8 9 10
  - i = 1, j = 2
1 2 3 ~~4~~ 5 6 ~~7~~ 8 9 10
  - i = 1, j = 3
1 2 3 ~~4~~ 5 6 ~~7~~ 8 9 ~~10~~
  - primes:
    -
    -
    -
    -
    -
    -
    -
    -