CSCI 2270
Data Structures and Algorithms
Lecture 22

Elizabeth White
elizabeth.white@colorado.edu
Office hours: ECCS 128
Wed 1-2pm
Fri 2-3pm

Administrivia

Labs 3, 4 will be graded soon; we're launching the COG autograder.

But there might be some bugs. This is new this semester.

Lab this week: big_number part 1

And yes, your add_node function is biased to sort

Extended part 1 deadline to Saturday at 11:55 pm

Additional bugs(!) in doubly linked list code: posted correction

Today: More big_number functions

Moving into other bases

Int constructor: no change, base remains 10

String constructor: check if digit > maximum, update base

private string alphabet =
"0123456789abcdefghijklmnopqrstuvwxyz";
finding the position of a letter in this string gives you the digit value for this letter

Copy constructor: no change except for updating base

Assignment operator: no change except for updating base

Additions

Additions (base 10)

999 + 1

999 + -1

-999 + 1

-999 + -1

1 + 999

-1 + 999

1 + -999

-1 + -999

Carrying…

Subtractions

Subtractions (base 10)

1000 - 1

1000 - -1

-1000 - 1

-1000 - -1

1 - 1000

-1 - 1000

1 - -1000

-1 - -1000

Borrowing…

Multiplications

234 * 678 = 234 * 8 + 234 * 70 + 234 * 600

Digit by digit multiplication is needed for speed (don’t just add!)

Converting into other bases

Convert 1024 in base 10 to number m in base 2

Begin by expressing 0 to 10 in binary:

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Converting into other bases

Convert 1024 in base 10 to number m in base 2

Start with m = 0

for each digit d in the old base k, highest to lowest:

m = m * k

m = m + d

1: m = 0 * 1010

m = 0 + 1

0: m = 1 * 1010 = 1010

m = 1010 + 0 = 1010

2: m = 1010 * 1010 = 1100100

m = 1100100 + 10 = 1100110

4: m = 1100110 * 1010 = 1111111100

m = 1111111100 + 100 = 10000000000

Converting into other bases

Convert 1024 in base 10 to number m in base 2

Start with m = 0

for each digit d in the old base k, highest to lowest:

m = m * k

m = m + d

1: m = 0 * 1010

m = 0 + 1

0: m = 1 * 1010 = 1010

m = 1010 + 0 = 1010

2: m = 1010 * 1010 = 1100100

m = 1100100 + 10 = 1100110

4: m = 1100110 * 1010 = 1111111100

m = 1111111100 + 100 = 10000000000

Converting into other bases

Convert 10000000000 in base 2 to number m in base 10

Start with m = 0

for each digit d in the old base k, highest to lowest:

m = m * k

m = m + d

1: m = 0 * 2 = 0

m = 0 + 1 = 1

0: m = 1 * 2 = 2

m = 2 + 0 = 2

0: m = 2 * 2 = 4

m = 4 + 0 = 4

// m doubles 8 more times: 8, 16, 32, 64, 128, 256, 512, 1024!