CSE 101 3-2-21

long has range:

$$-2^{63} \le x \le 2^{63} - 1$$

base 10 : $[357]_{10}$

$$[357]_{10} = 3 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0$$

base b : $[c_{n-1}\, c_{n-2} \cdots c_2\, c_1\, c_0]_b$

$$= c_{n-1} \cdot b^{n-1} + c_{n-2} \cdot b^{n-2} + \cdots + c_2 b^2 + c_1 b^1 + c_0 b^0$$

digits

range: $0 \le c_i \le b-1$

Ex $b = 10^2 = 100$

$$digits = \{00, 01, 02, \ldots, 97, 98, 99\}$$

$$[2743]_{10} = [02, 73, 43]_{100}$$

check

$$(02)\cdot 100^2 + (73)100^1 + (43)\cdot 100^0$$

$$= 2\cdot 10000 + 73\cdot 100 + 43$$

$$= 20000 + 7300 + 43 = [27543]_{10}$$

We take $b = 10^p$ where

$$1 \le p \le 9$$

base 10 — base 1000000000

Example $b=100$, $p=2$

$$472038 = (47\ 20\ 38)$$
$$232509 = (23\ 25\ 09)$$

$$704547 = (70\ 45\ 47)$$

Ex.

$$\begin{array}{r} {\scriptstyle 1\quad\ 1} \\ 35\ 57\ 97 \\ 14\ 90\ 82 \\ \hline 50\ 48\ 79 \end{array}$$

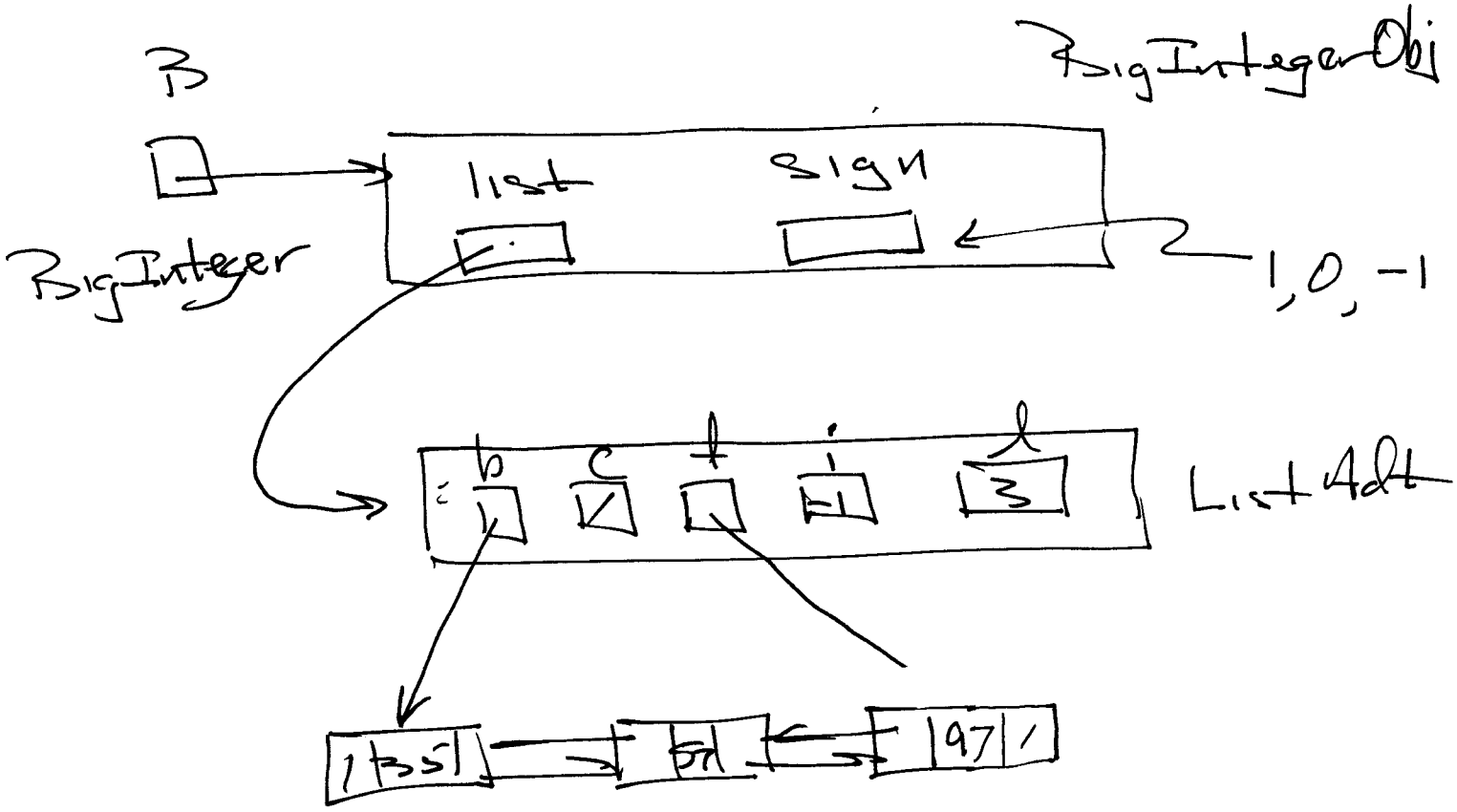
another way: add as vectors

$$\begin{array}{r} (35, 57, 97) \\ (14, 90, 82) \\ \hline {\scriptstyle 1\quad\ 1} \\ (49, 147, 179) \\ {\scriptstyle -100\ -100} \end{array}$$

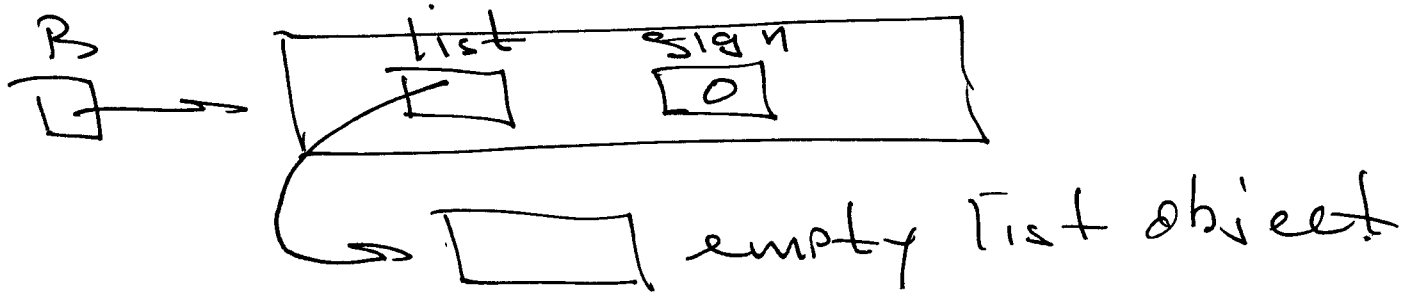
normalize $(50, 48, 79)$

Advice: work on fcns in this order

- constructor & destructor.



Zero state



note:

ignore: divide(), quot(), rem().

next build fcns

- stringToBigInteger()
- printBigInteger()
- compare(), equal()

helper fcns:

- normalize()
- negate()
- addLists()
- scalarMultiplyLists()

Then:

add()
sub()
mult()

Heading for normalizeList()

int normalizeList(List L);

Back to Priority Queues ...

why do we have them?

Chap. 24 SSSP in weighted graphs

Definitions

- weighted graph: $G = (V, E)$ with a weight fcn

$$w : E \longrightarrow \mathbb{R}$$

Pos, neg, or zero.

- weight of a subgraph : $H \subseteq G$

$$V(H) \subseteq V(G)$$

$$E(H) \subseteq E(G)$$

$$w(H) = \sum_{e \in E(H)} w(e)$$

<u>special case</u> : an x-y Path P

$P : x = v_0, v_1, \ldots, v_k = y$ { no repeated edges or vertices.

$$w(P) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

- min weight Path :

$w(P) \leq w(q)$ { for all x-y Paths q

↑

min weight Path

- min Path weight

$$\delta(x,y) = \begin{cases} w(P) : P \text{ is a min weight } x\text{-}y \text{ Path} \\ \infty \quad \text{no } x\text{-}y \text{ Path exists} \end{cases}$$

- SSSP Problem.

given $s \in V(G)$, determine $\delta(s,x)$ for all $x \in V(G)$. for those x with $\delta(s,x) < \infty$, find an s-x Path P with $w(P) = \delta(s,x)$.

Two Algorithms:

Dijkstra

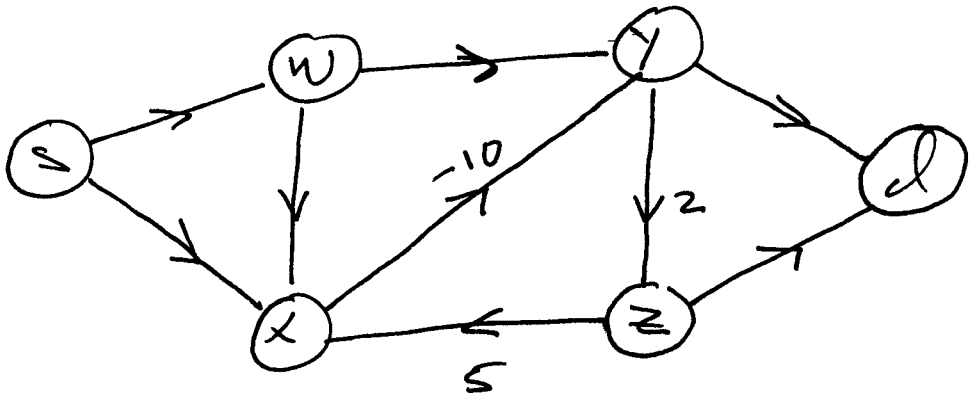
Bellman-Ford

Both actually find min weight walks in G. (works because a min weight walk is necessarily a min weight path.)

If a graph has neg. weight edges, then there may be no min weight walk. In Particular if there is a min. weight cycle reachable from source S.

Ex.

G



$w(x, y, z, x) = -3$

In this case there is no min-weight s-d walk in G.

Dijkstra: only works on graphs with non-negative edge weights.

Bellman-Ford. detects existence of neg. weight cycles reachable from source.