class Node (var datum : Int) {

var nexts = new Array[Node](32)

}

class SkipList {

// Initialisation

private var adds = 0

var tail = new Array[Node](32)

for (i <- 0 until 32) tail(i) = new Node(Int.MaxValue)

var head = new Array[Node](32)

for (i <- 0 until 32) {head(i) = new Node(Int.MinValue) ; head(i).nexts = tail}

def listAtLevel (level : Int) : Array[Node] = {

var a = new Array[Node](100)

var k = 0

var current = head(level).nexts(level)

while (current.datum != Int.MaxValue)

{

a(k) = current

k += 1

current = current.nexts(level)

}

a(k) = new Node(Int.MaxValue)

a

}

def listAtLevelInt (level : Int) : Array[Int] = {

var a = new Array[Int](100)

var b = listAtLevel(level)

var i = 0

while (b(i) != null) {a(i) = b(i).datum ; i += 1}

a

}

def findPreds (x : Int) : Array[Node] = {

var preds = new Array[Node](32)

var current = head(31)

var down = head(30)

var i = 31

while (i > 0)

{

while ((current.nexts(i) != null) && (current.nexts(i).datum < x))

{

down = current.nexts(i-1)

current = current.nexts(i)

}

preds(i) = current

current = down

i -= 1

}

// i = 0

while ((current.nexts(0) != null) && (current.nexts(0).datum < x))

current = current.nexts(0)

preds(0) = current

preds

}

def findPredsInt (x : Int) : Array[Int] = {

var a = new Array[Int](100)

var b = findPreds(x)

var i = 0

while ((i < 2) && (b(i) != null)) {a(i) = b(i).datum ; i += 1}

a

}

def contains (x : Int) : Boolean =

((findPreds(x)(0).nexts(0) != null) && (findPreds(x)(0).nexts(0).datum == x))

def add (x : Int) = {

adds += 1

var n = pickLevel

var preds = findPreds(x)

var i = n

var a = new Array[Node](32)

for (i <- 0 to n) a(i) = preds(i).nexts(i)

while (i >= 0)

{

var n1 = new Node(x)

preds(i).nexts(i) = n1

preds(i).nexts(i).nexts = a

i -= 1

}

}

def pickLevel : Int = {

var i = 0

var x = 1

while (adds % x == 0) {x = x \* 2 ; i += 1}

i-1

}

def delete (x : Int) = if (contains(x)) {

var preds = findPreds(x)

var i = 0

while ((i<32) && (preds(i) != null))

{

if (preds(i).nexts(i) != null) preds(i).nexts = preds(i).nexts(i).nexts

i += 1

}

}

}