```lua
local _,the,SYM = require"lib", require"the", require"sym"
local fmt,per,upx,push,sort = _.fmt,_.per,_.upx,_.push,_.sort
local ent,o,oo = _.ent,_.o, _.oo
local class,OBJ = _.class, _.OBJ

local BIN=class("BIN",OBJ)
function BIN:new(at,name, lo,hi,ys)
  self.at, self.name        = at or 0, name or ""
  self.lo, self.hi, self.ys = lo, hi or lo, ys or SYM() end

function BIN:__tostring()
  local x,lo,hi,big = self.name, self.lo, self.hi, math.huge
  if      lo ==  hi  then return fmt("%s==%s",x, lo)
  elseif hi ==  big then return fmt("%s>=%s",x, lo)
  elseif lo == -big then return fmt("%s<%s",x, hi)
  else              return fmt("%s<=%s < %s",lo,x,hi)  end end

function BIN:select(row)
  local x, lo, hi = row[self.at], self.lo, self.hi
  return x=="?" or lo == hi and lo == x or lo <= x and x < hi end

function BIN:add(x,y)
  if x<self.lo then self.lo = x end
  if x>self.hi then self.hi = x end
  self.ys:add(y) end


function BIN.mergeSameDivs(b4,after)
  local merged = b4.ys:merged(after.ys)
  if merged then
    return BIN(b4.at, b4.name, b4.lo, after.hi, merged) end end

function BIN.mergeNext(b4,after)
  if b4.hi == after.lo and b4.lo ~= b4.hi then
    return BIN(b4.at, b4.name, b4.lo, after.hi, b4.ys:merge(after.ys)) end end

return BIN
```