GetNode

The following function allows navigating a vEB explicitly saved as a tree with pointers (not a heap), returning a pointer to the i^{th} node.

The intuition is to recursively split the tree in two by halving its height and navigating to the leaves of the subtrees thus defined that lie on the path to the destination.

```
GetNode(curr, i, h) /* curr root of tree, i-th node to reach, h height of tree */
  if (h <= 2) then
    /* recursion endcase. Note that it takes care of singleton trees,
      assuming a sensible query (i.e. h = 1 => i = 0) */
    if (i = 0) then
      return curr;
    elif (i = 1) then
      return curr.L;
    else
      return curr.R;
    end
  end
  let zs = ZoomSequence(i, h);
  let b = zs.Dequeue();
  if (b = 0) then
    /* path to destination stays in the top tree */
    return GetNode(curr, i mod 2^(h/2 -1), h/2);
  end
  /* path to destination reaches a bottom tree so let's traverse the top tree */
  curr = ReachLeaf(curr, (b-1)/2, h/4); /* navigate first half of top tree */
  curr = ReachLeaf(curr, b mod 2^(h/4 -1), h/4); /* navigate second half of top tree */
  /* we're on a leaf of the top tree, so let's reach the appropriate bottom tree */
  if (b-1 mod 2 = 0) then
    curr = curr.L;
  else
    curr = curr.R;
  end
  return GetNode(curr, i mod 2^(h/2 -1), h/2); /* recursion on bottom tree */
end
```

ReachLeaf

The following function returns a pointer to the i^{th} leaf of a non-singleton vEB. We number leaves starting from 0, left to right.

```
ReachLeaf(curr, l, h) /* curr root of tree, l-th leaf to reach, h height of tree */
  if (h = 2) then
    /* recursion endcase (we won't need to contemplate a singleton tree) */
    if (l = 0) then
      return curr.L;
    else
      return curr.R;
    end
  end
  curr = ReachLeaf(curr, l/2^(h-1), h/2);
  /* we have traversed the top half of the tree; now we must reach the bottom half.
    To do so, we look at the position of leaf l in a tree of height h+1 */
  if (l mod 2^h >= 2^(h/2)) then
    root = curr.R;
  else
    root = curr.L;
  end
  curr = ReachLeaf(curr, l mod 2^(h-1), h/2);
```

```
  return curr;
end
```