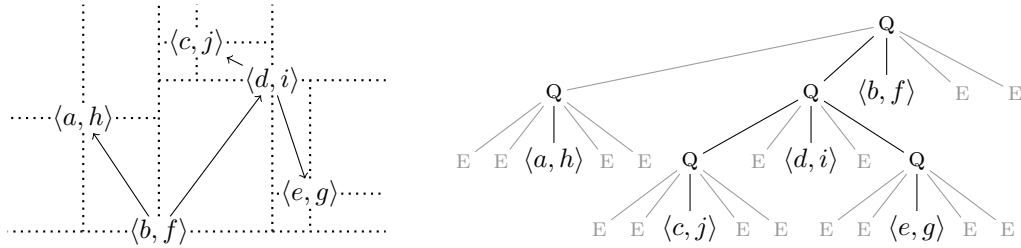


2 Foundations of Computer Science (jdy22)

This question considers the following type `qt` that represents *quadtrees* of points:

```
type point = int * int
type qt = Empty | Quad of qt * qt * point * qt * qt
```

A value `Quad(nw, ne, (x, y), sw, se)` contains points in the quadrants around (x, y) . x is the right bound of points in `nw` and `sw` and the left bound of points in `ne` and `se`. y is the upper bound of points in `sw` and `se` and lower bound of points in `nw` and `ne`.



- (a) Write a function `compare_range` to find whether a number falls below, within or above a range:

```
type range = int * int
type rel = LT | IN | GT
val compare_range : int -> range -> rel
```

For example, `compare_range 3 (2,5)` should return `IN`, because $2 \leq 3 \leq 5$.
[2 marks]

- (b) Write a function `has_point` to efficiently search a quadtree for a point:

```
val has_point : point -> qt -> bool
```

[8 marks]

- (c) Write a function `has_point_in` to efficiently search a quadtree for a point within a rectangular region:

```
type rectangle = point * point
val has_point_in : rectangle -> qt -> bool
```

`has_point_in (p1,p2) qt` should return `true` if and only if `qt` contains a point in the rectangular region with lower-left corner `p1` and upper-right corner `p2`.

[10 marks]