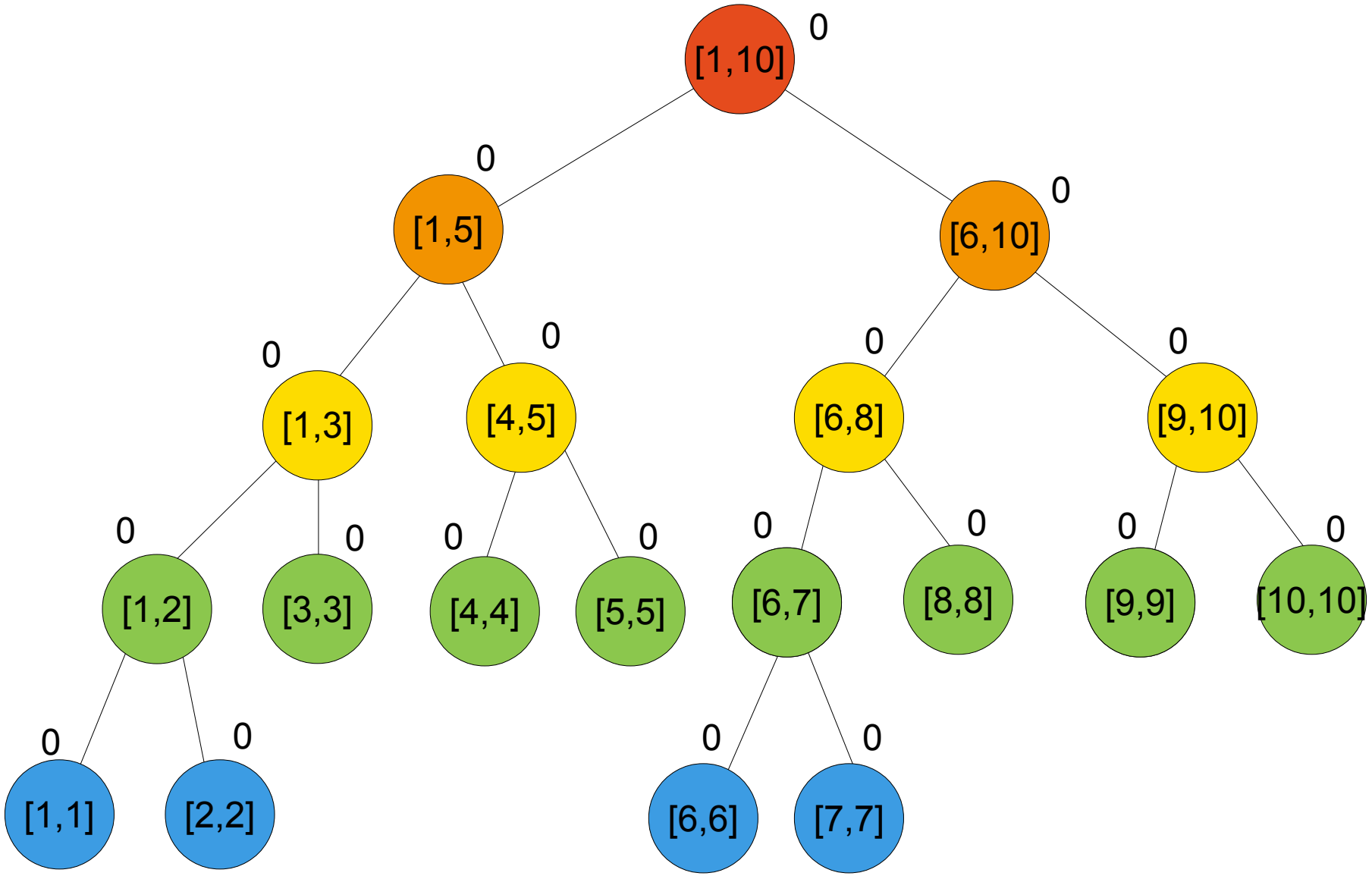
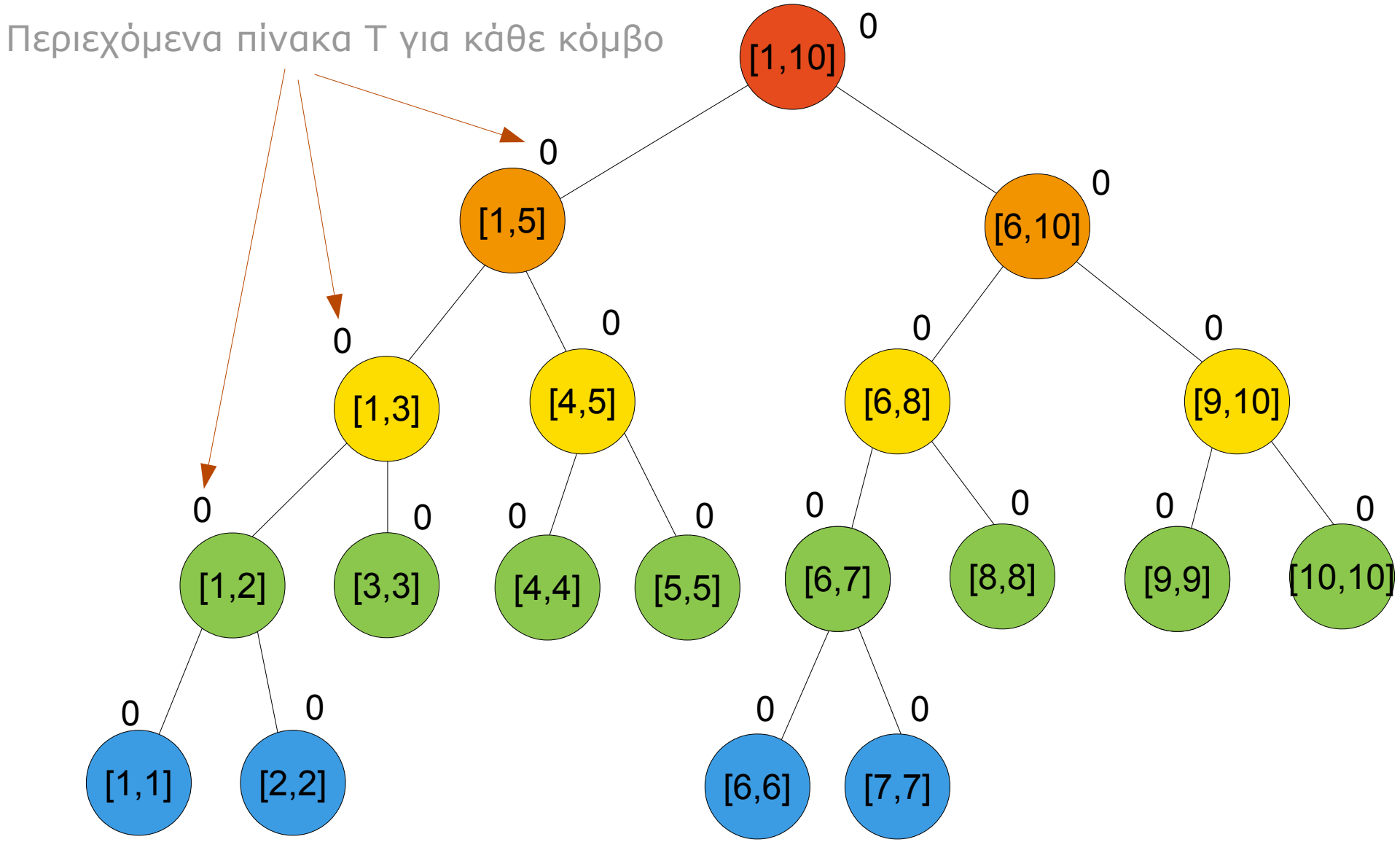
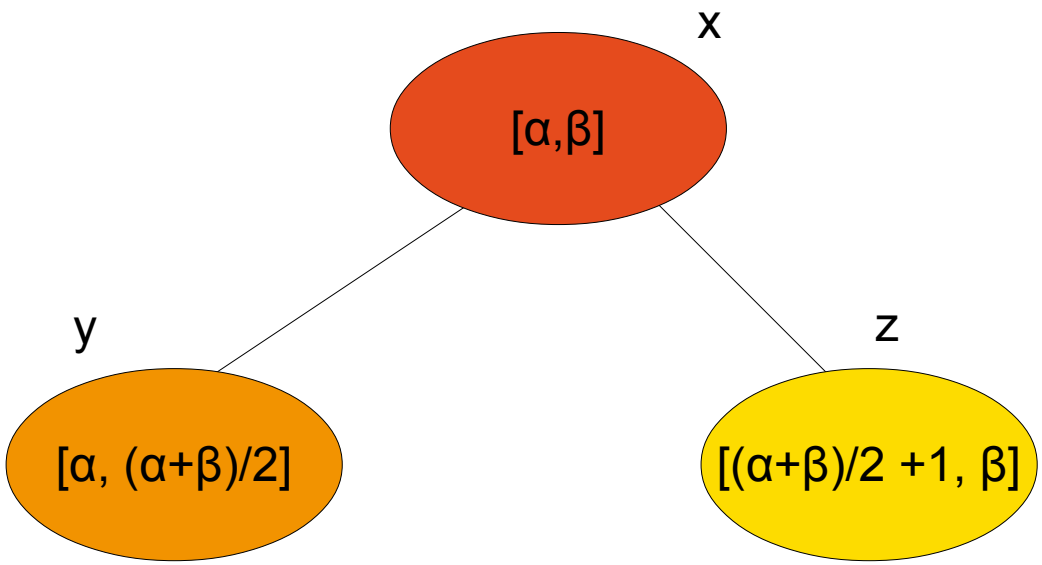
Interval Trees



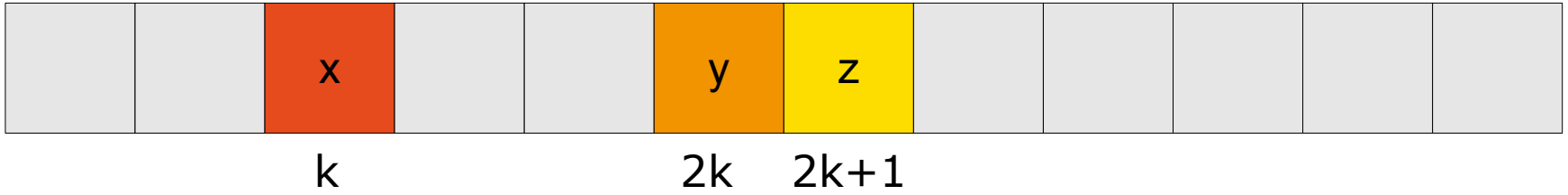
Interval Trees



Interval Trees (αναπαράσταση στη μνήμη)



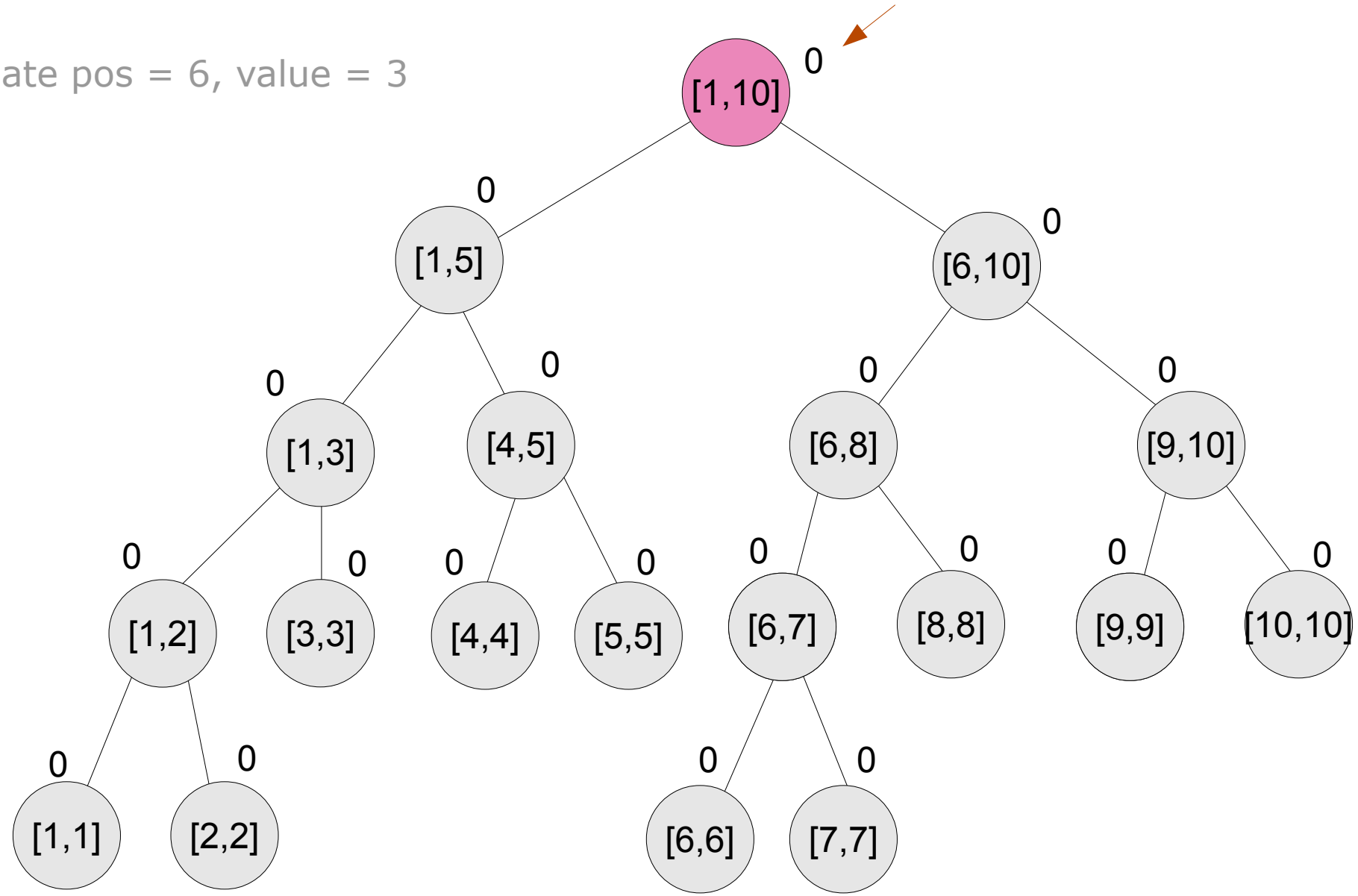
Πίνακας Τ:



Αριστερό παιδί του k : 2 * k
Δεξιό παιδί του k: 2 * k + 1
Πατέρας του k: k / 2

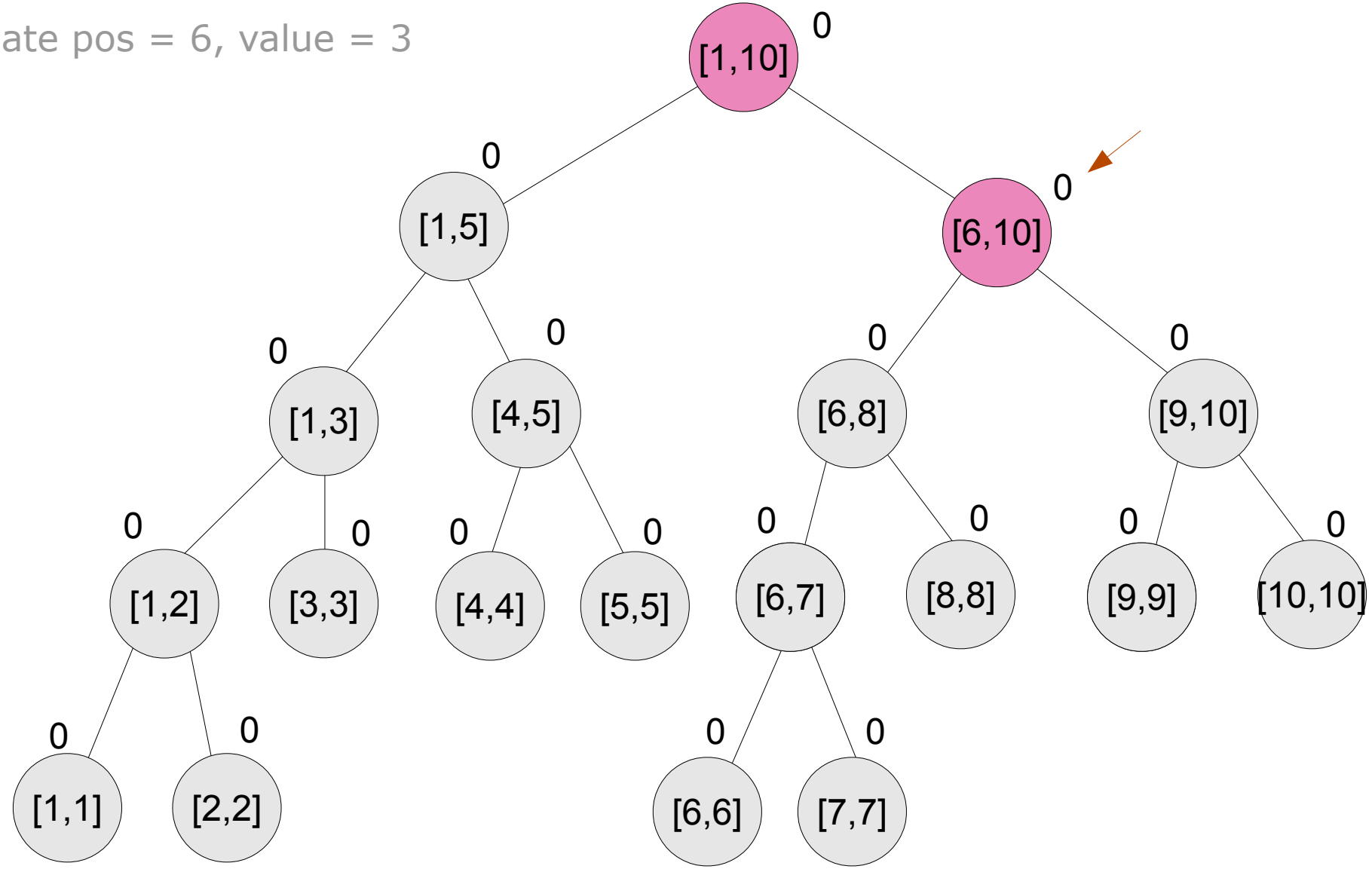
Interval Trees

Update pos = 6, value = 3



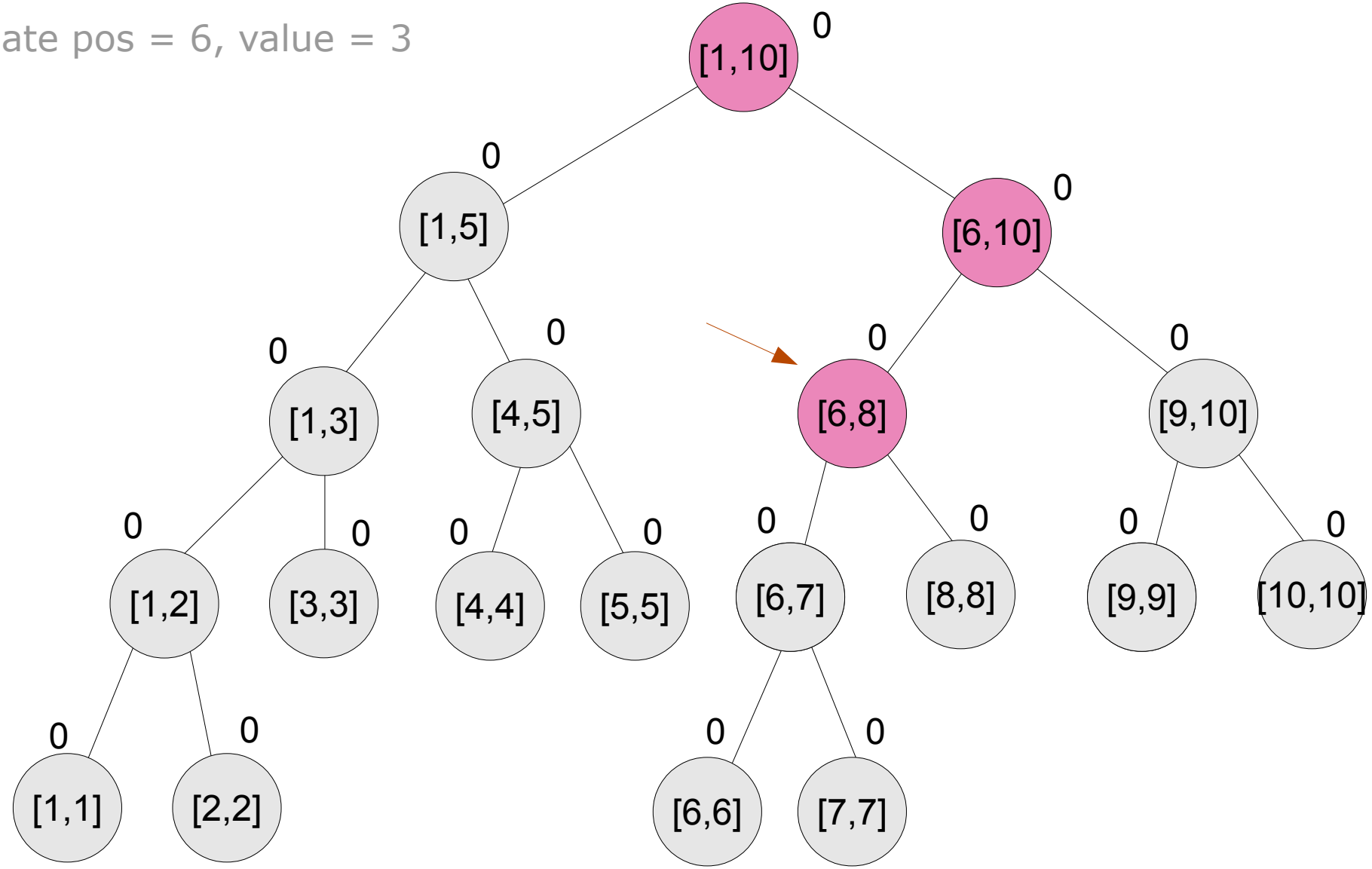
Interval Trees

Update pos = 6, value = 3



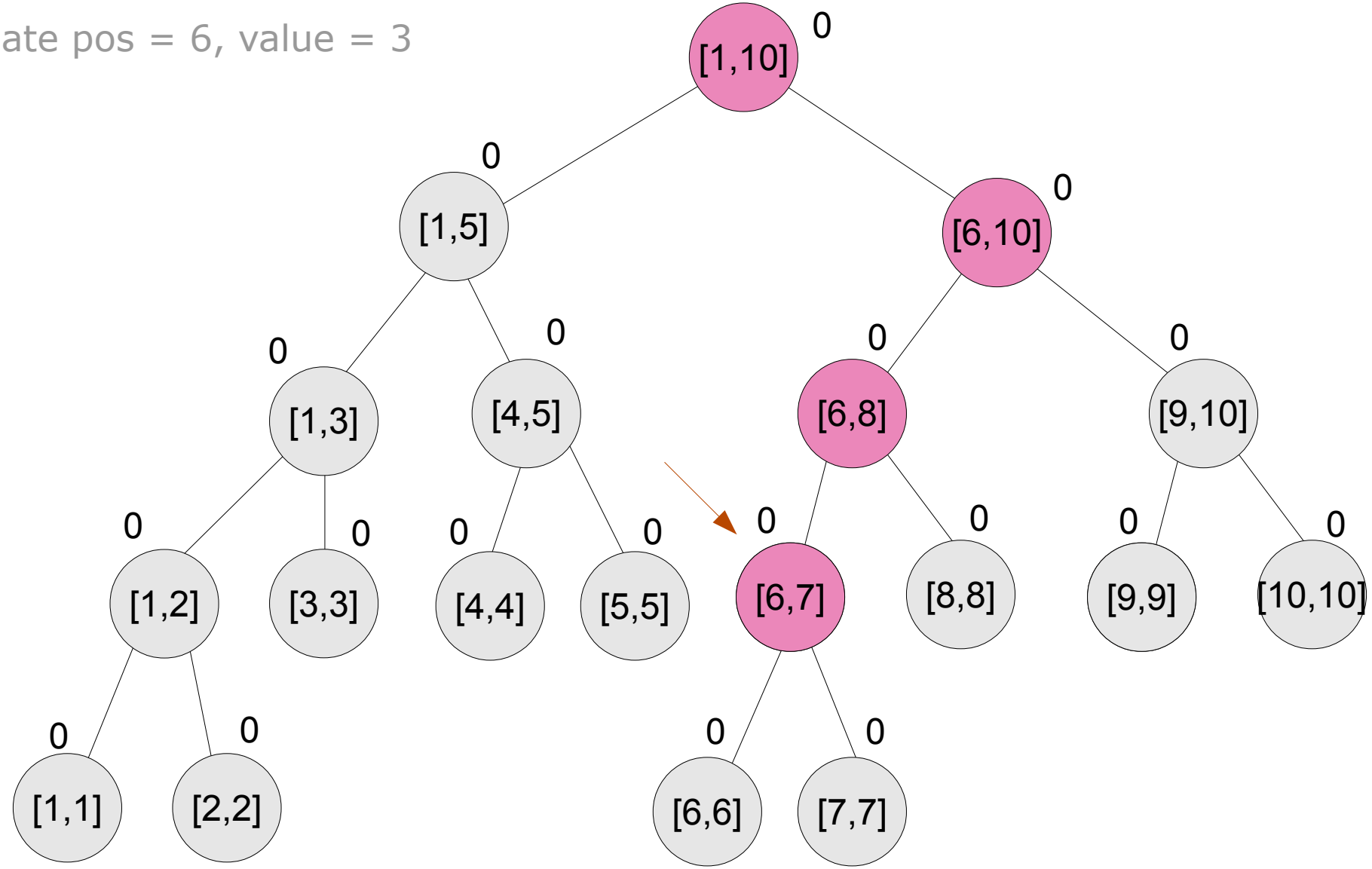
Interval Trees

Update pos = 6, value = 3



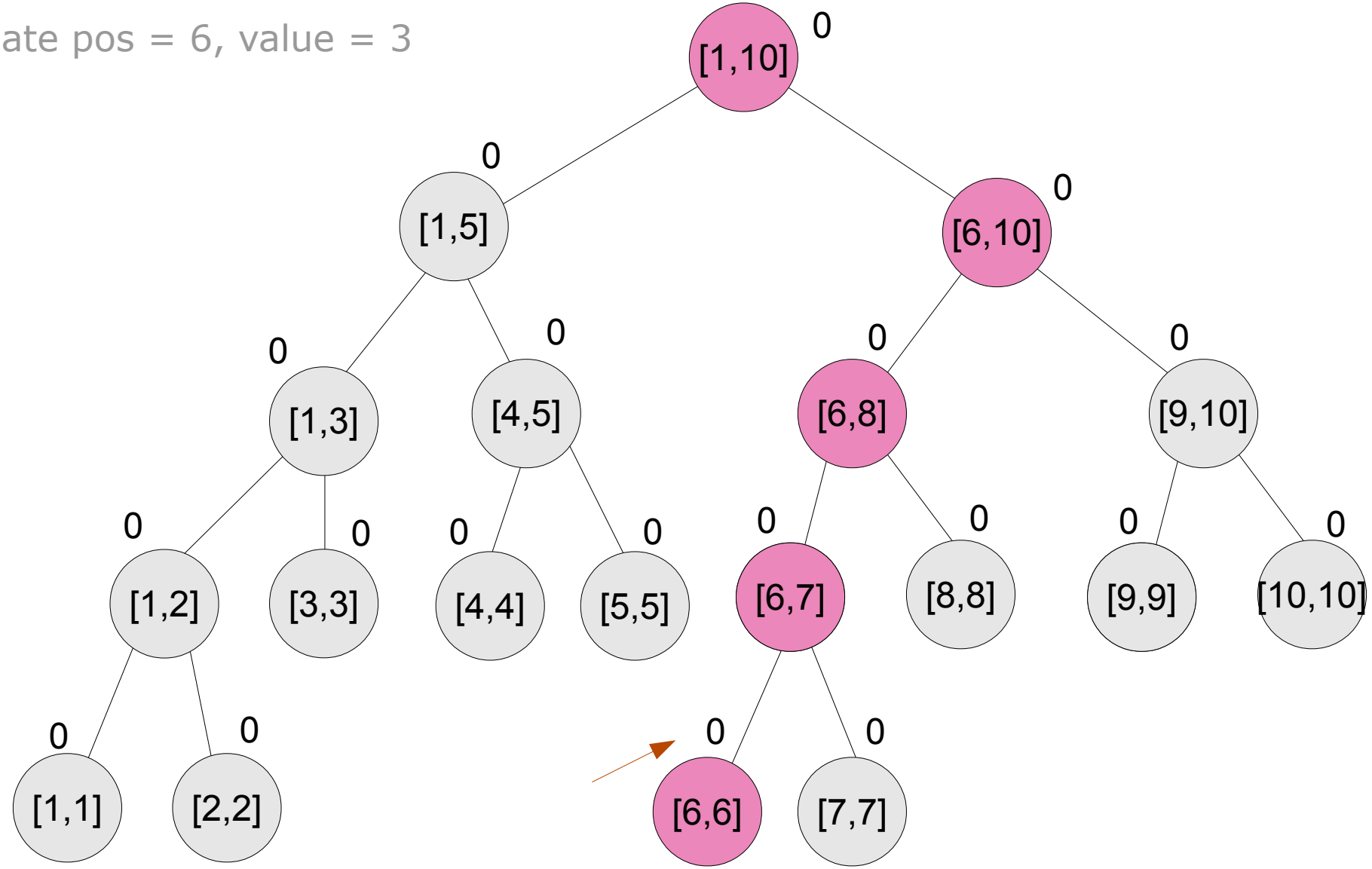
Interval Trees

Update pos = 6, value = 3



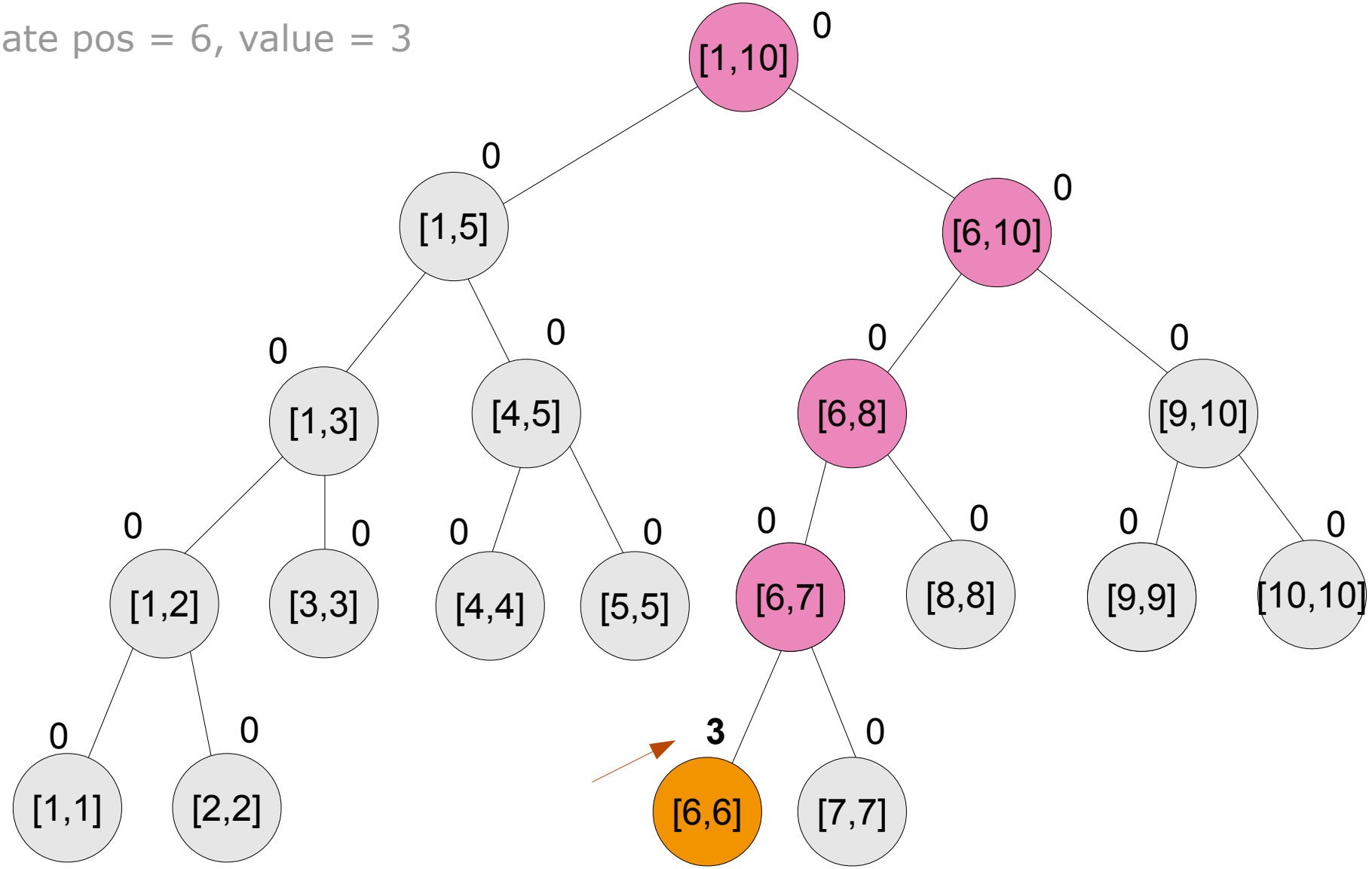
Interval Trees

Update pos = 6, value = 3



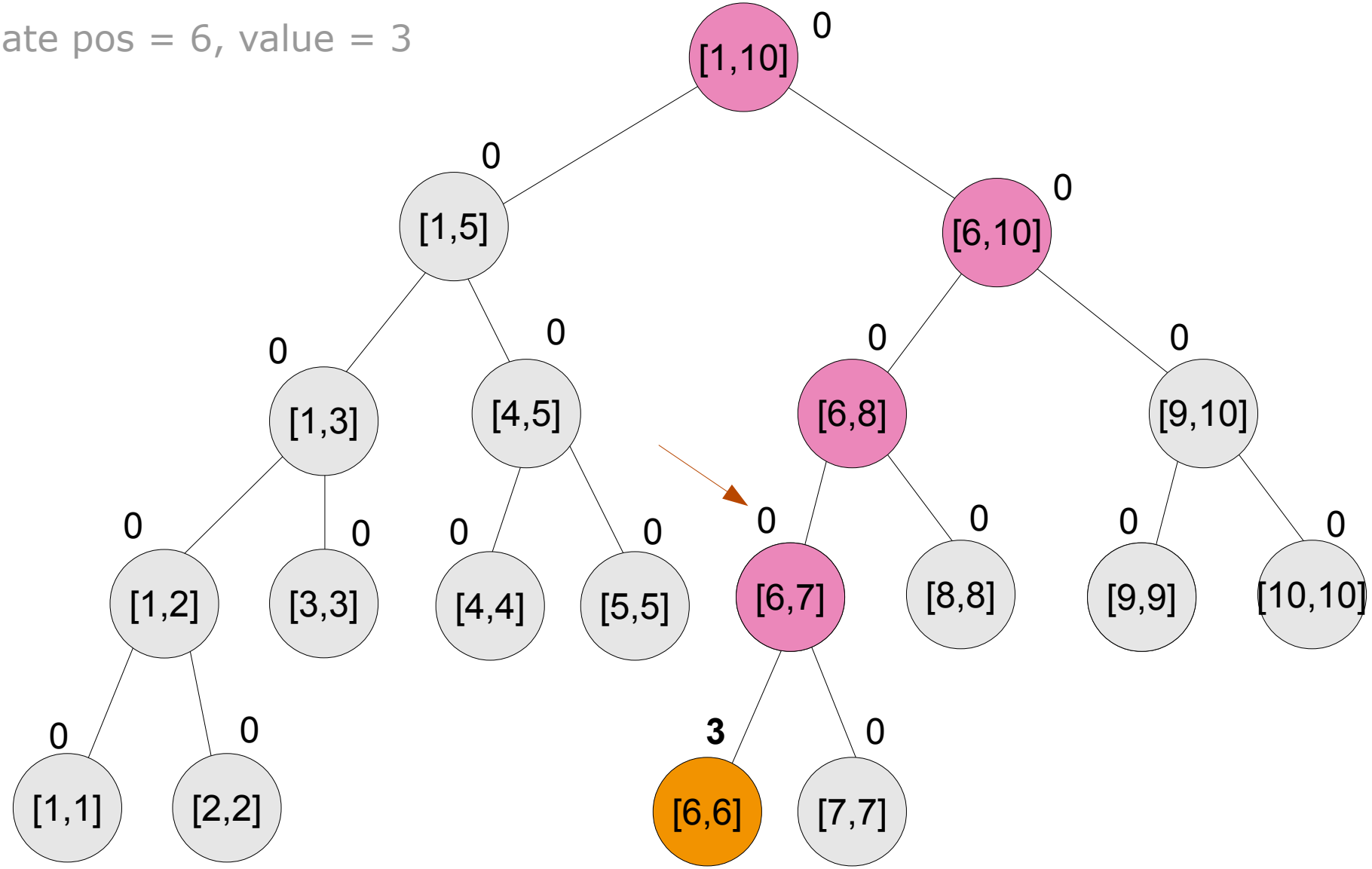
Interval Trees

Update pos = 6, value = 3



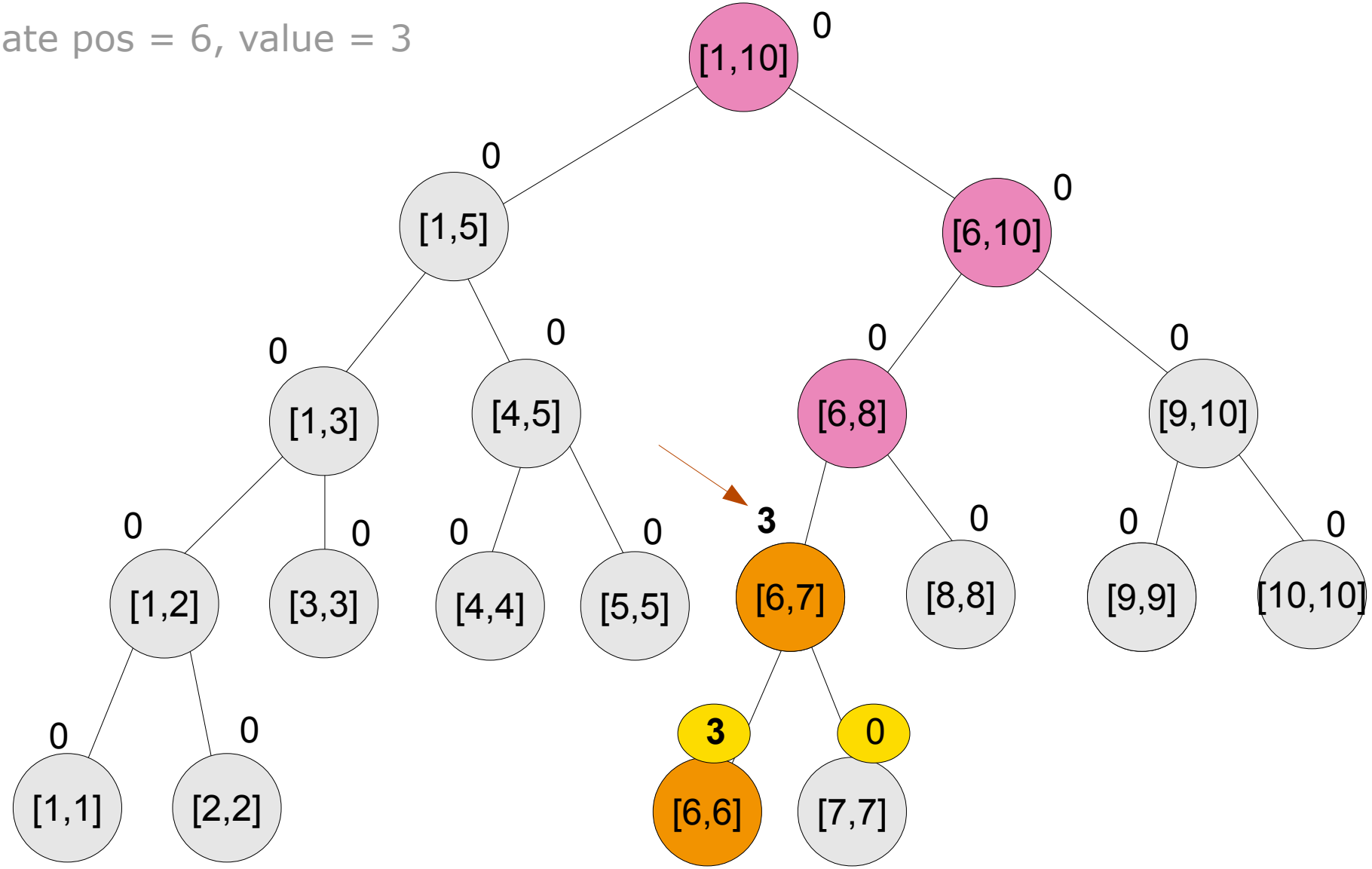
Interval Trees

Update pos = 6, value = 3



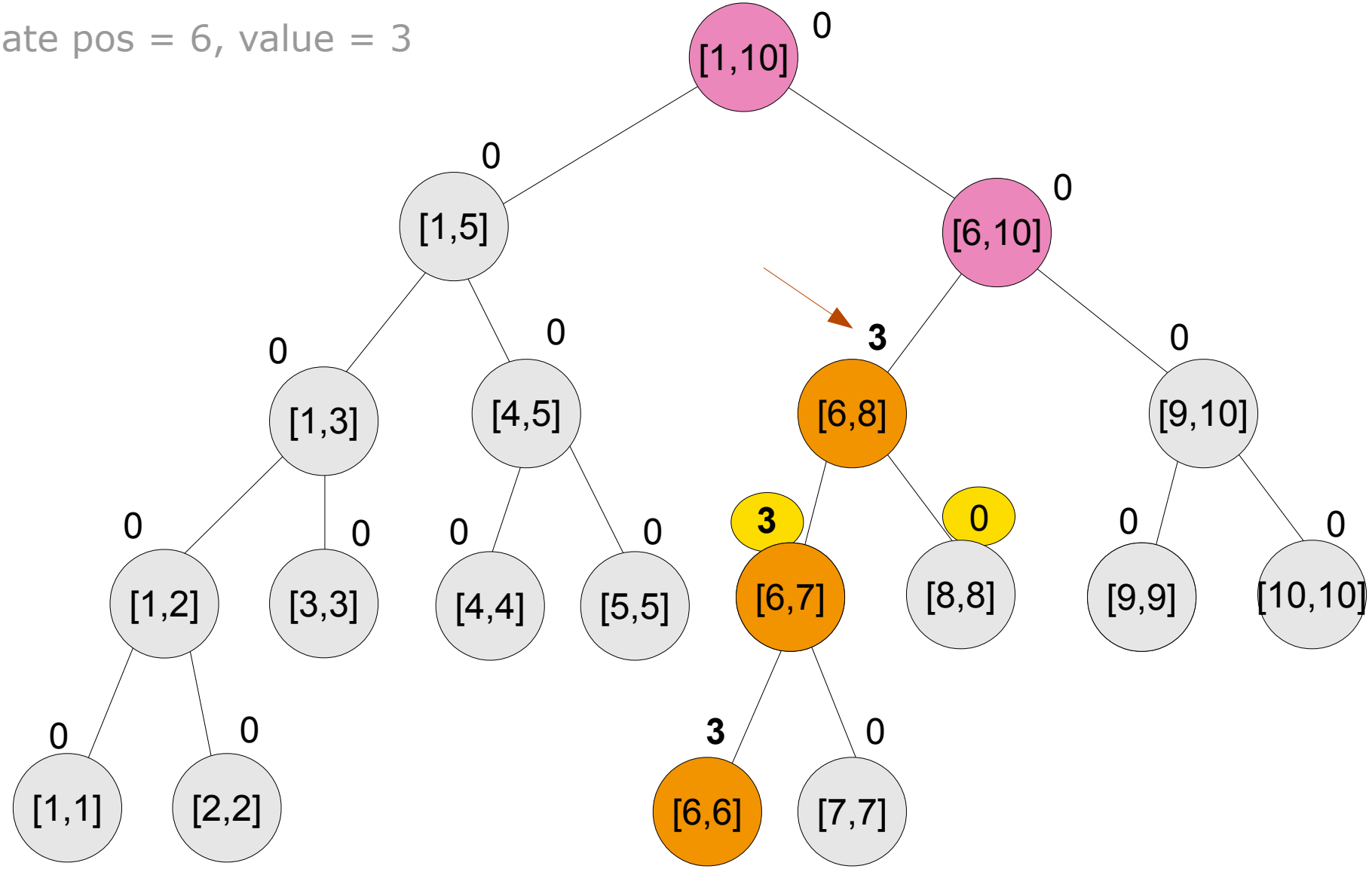
Interval Trees

Update pos = 6, value = 3



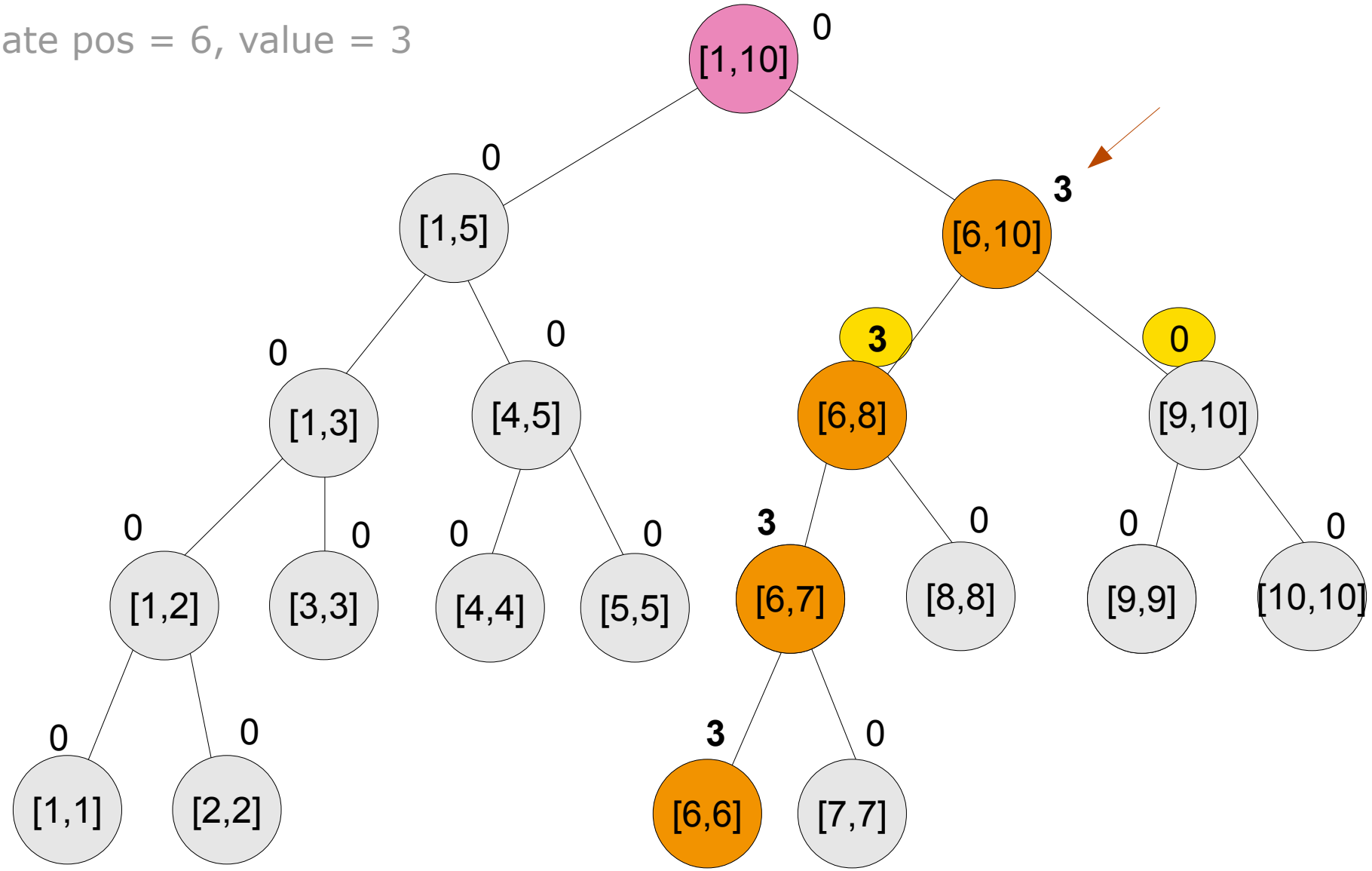
Interval Trees

Update pos = 6, value = 3



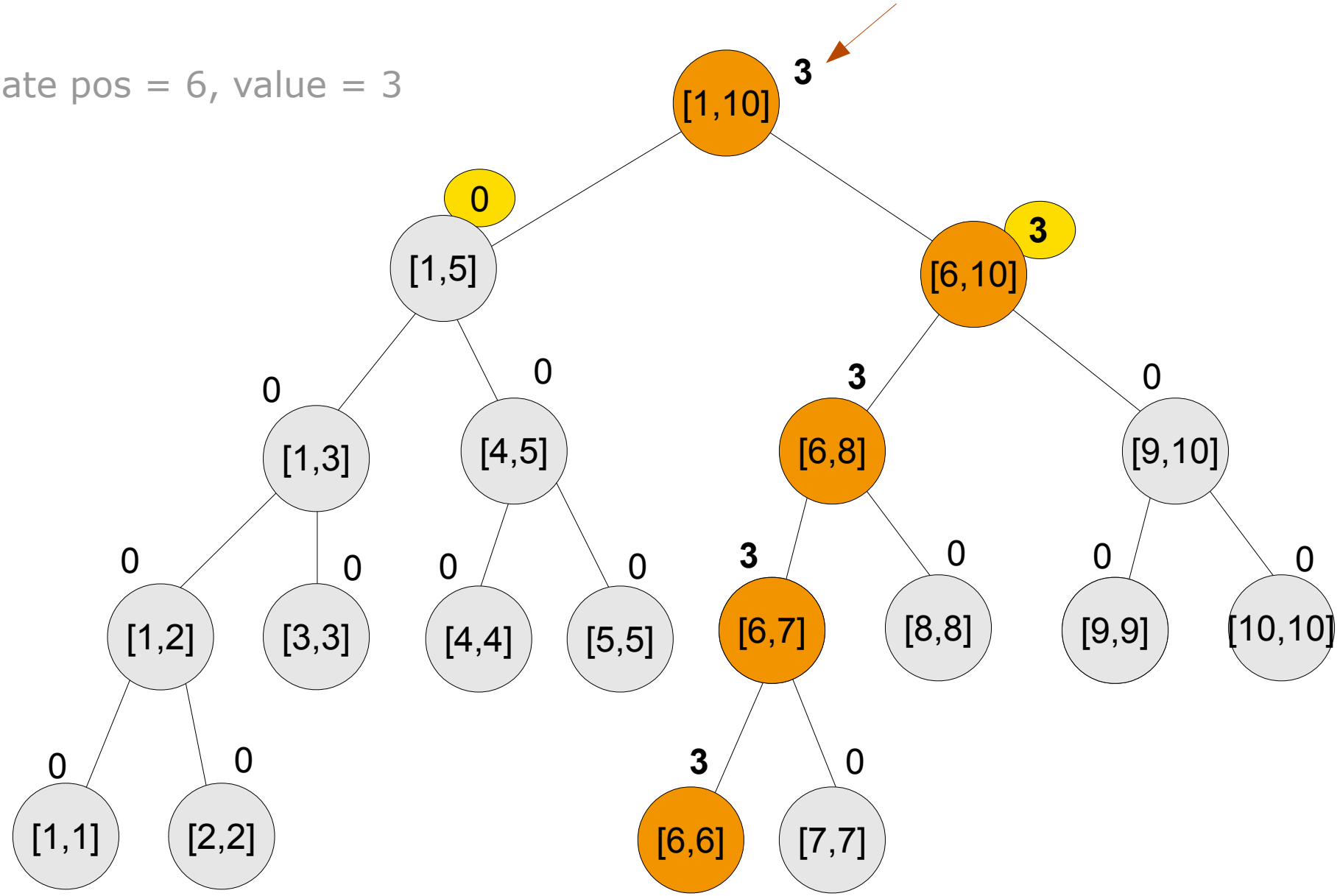
Interval Trees

Update pos = 6, value = 3



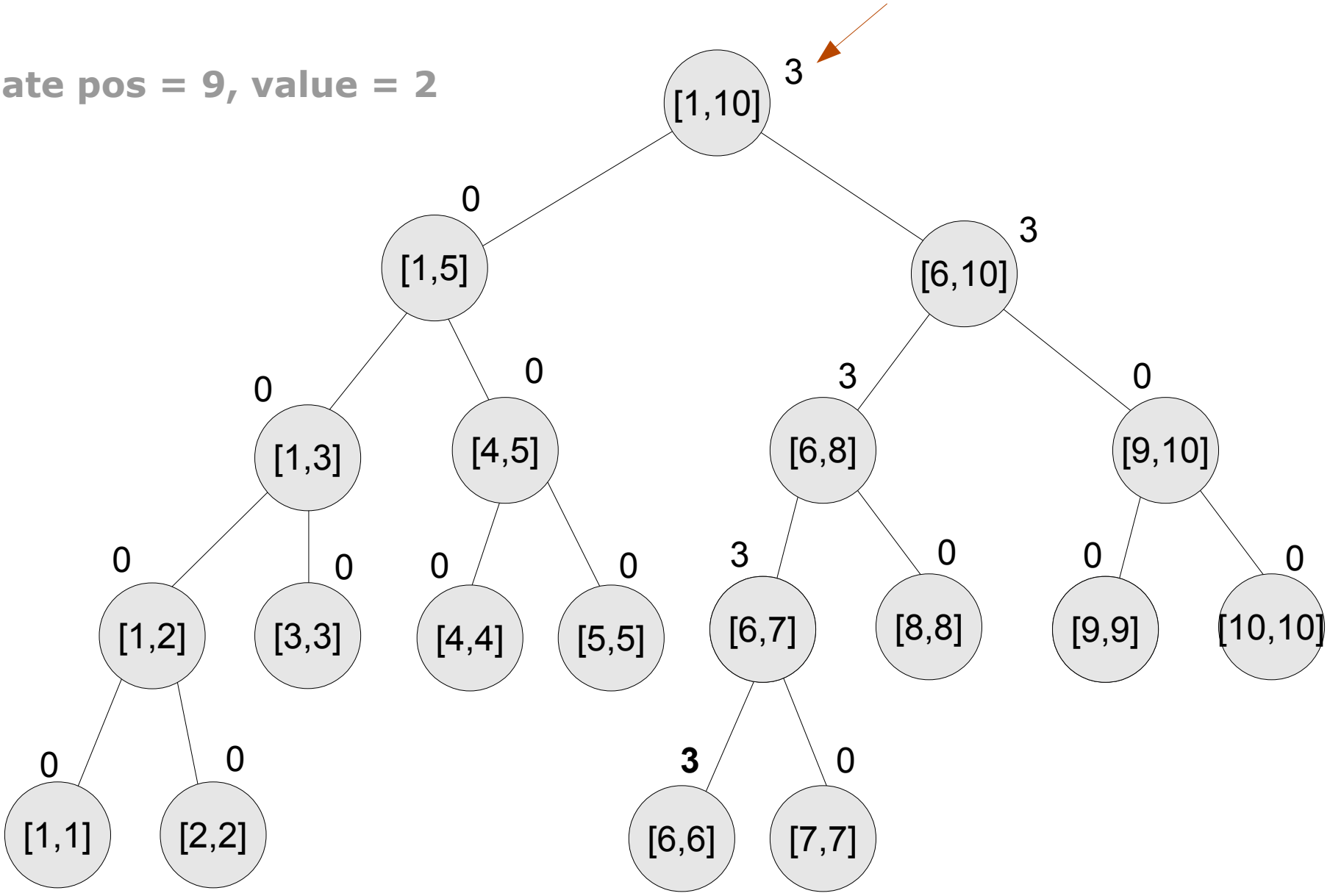
Interval Trees

Update pos = 6, value = 3



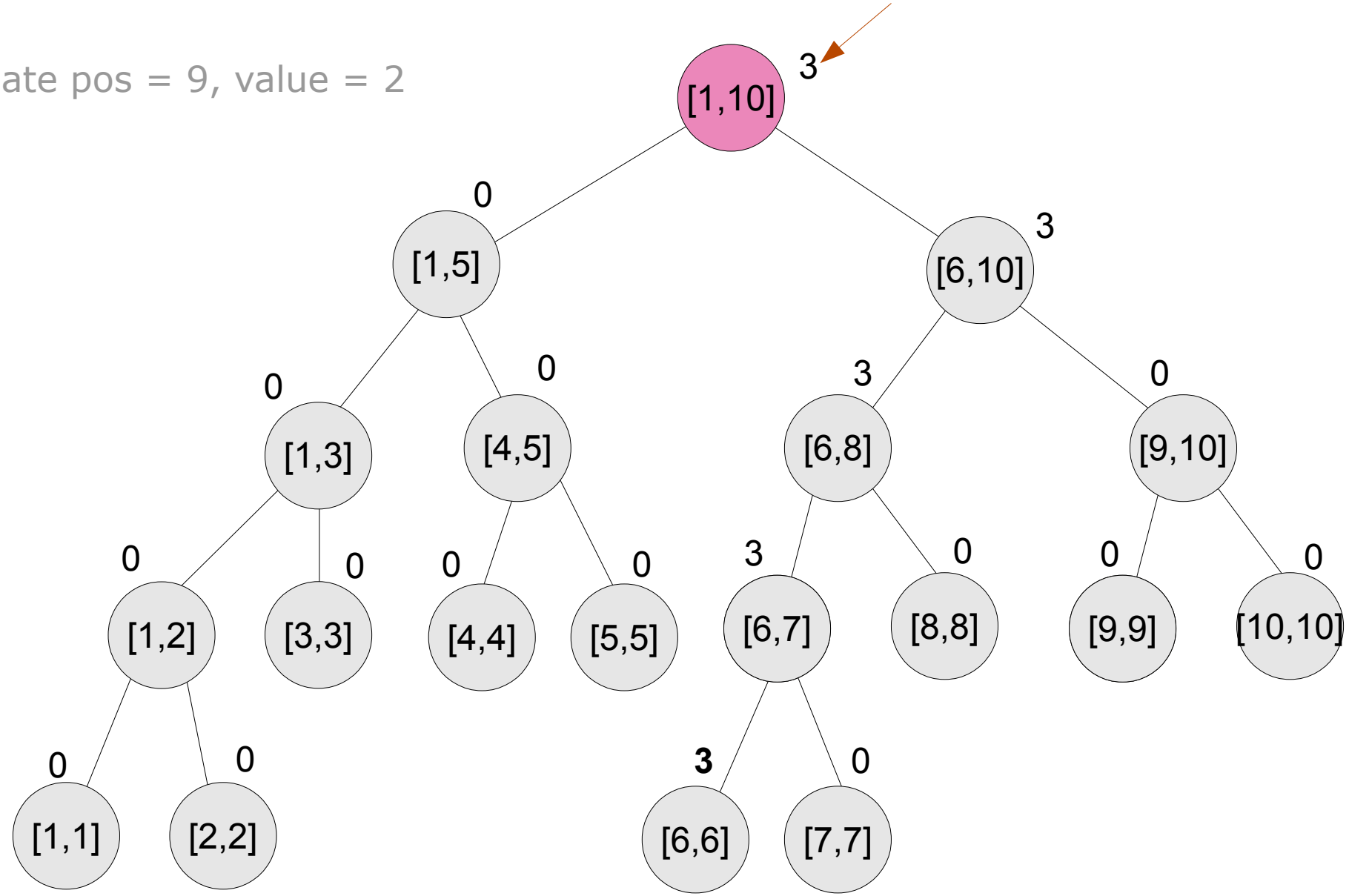
Interval Trees

Update pos = 9, value = 2



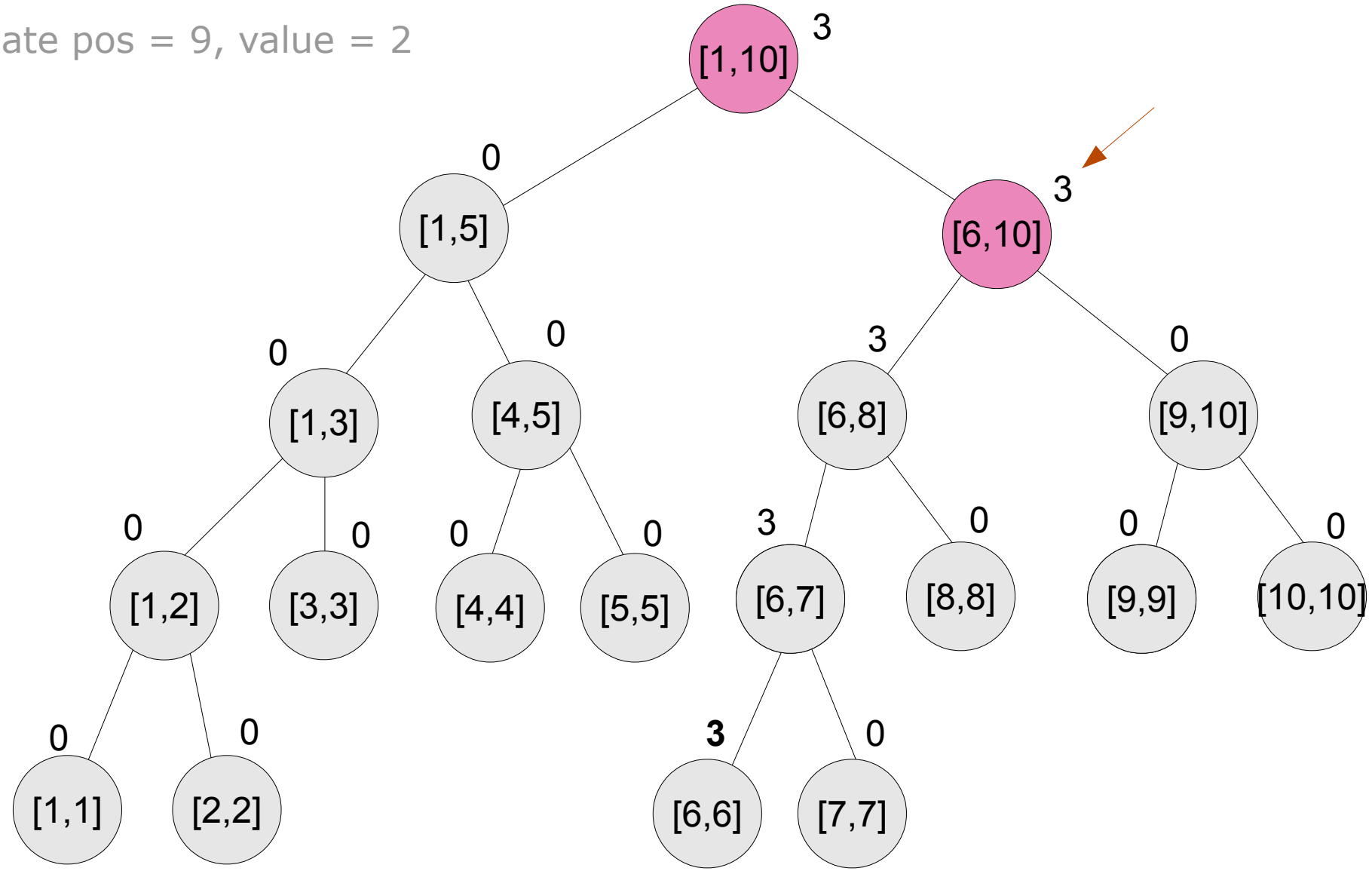
Interval Trees

Update pos = 9, value = 2



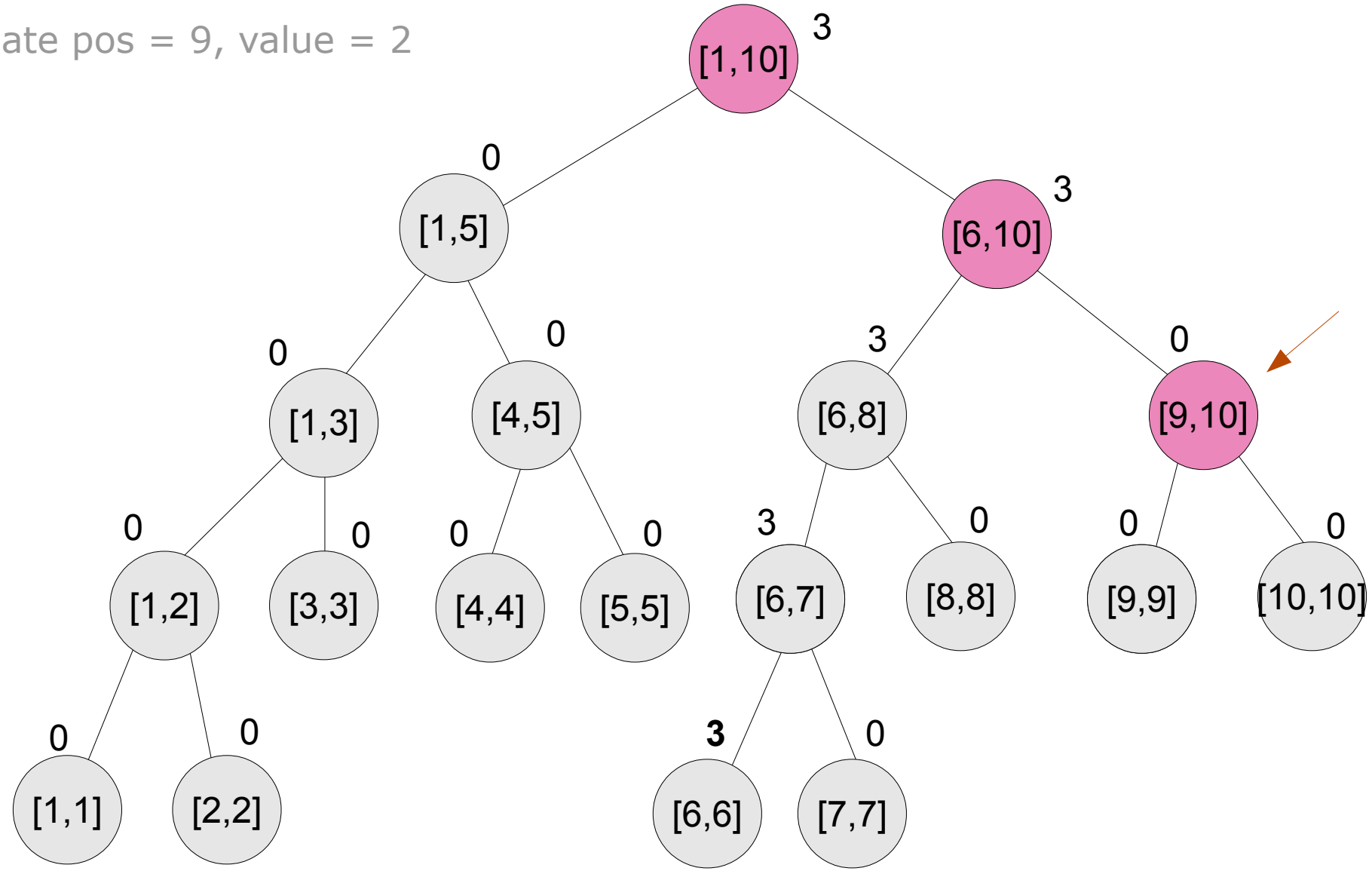
Interval Trees

Update pos = 9, value = 2



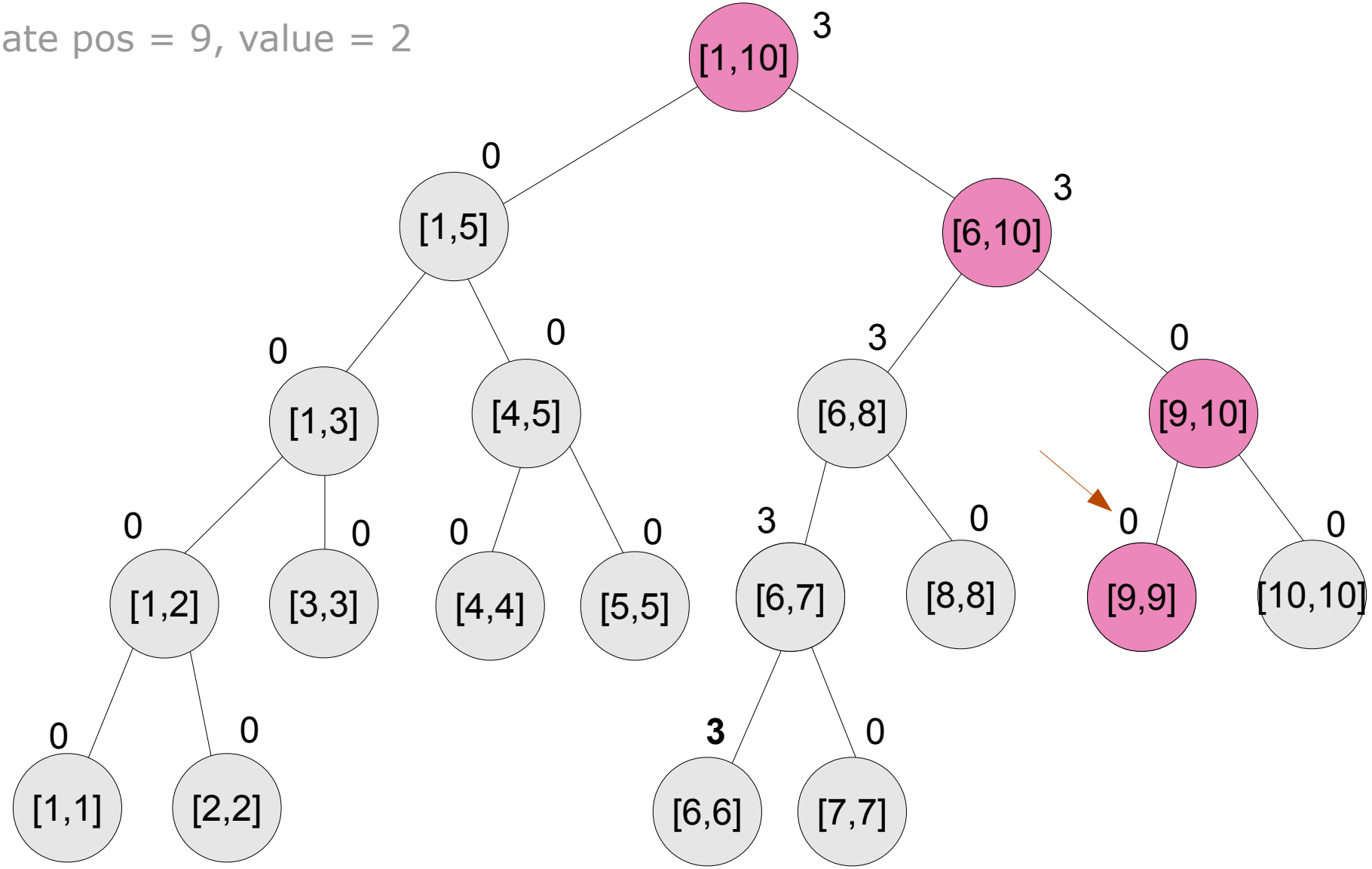
Interval Trees

Update pos = 9, value = 2



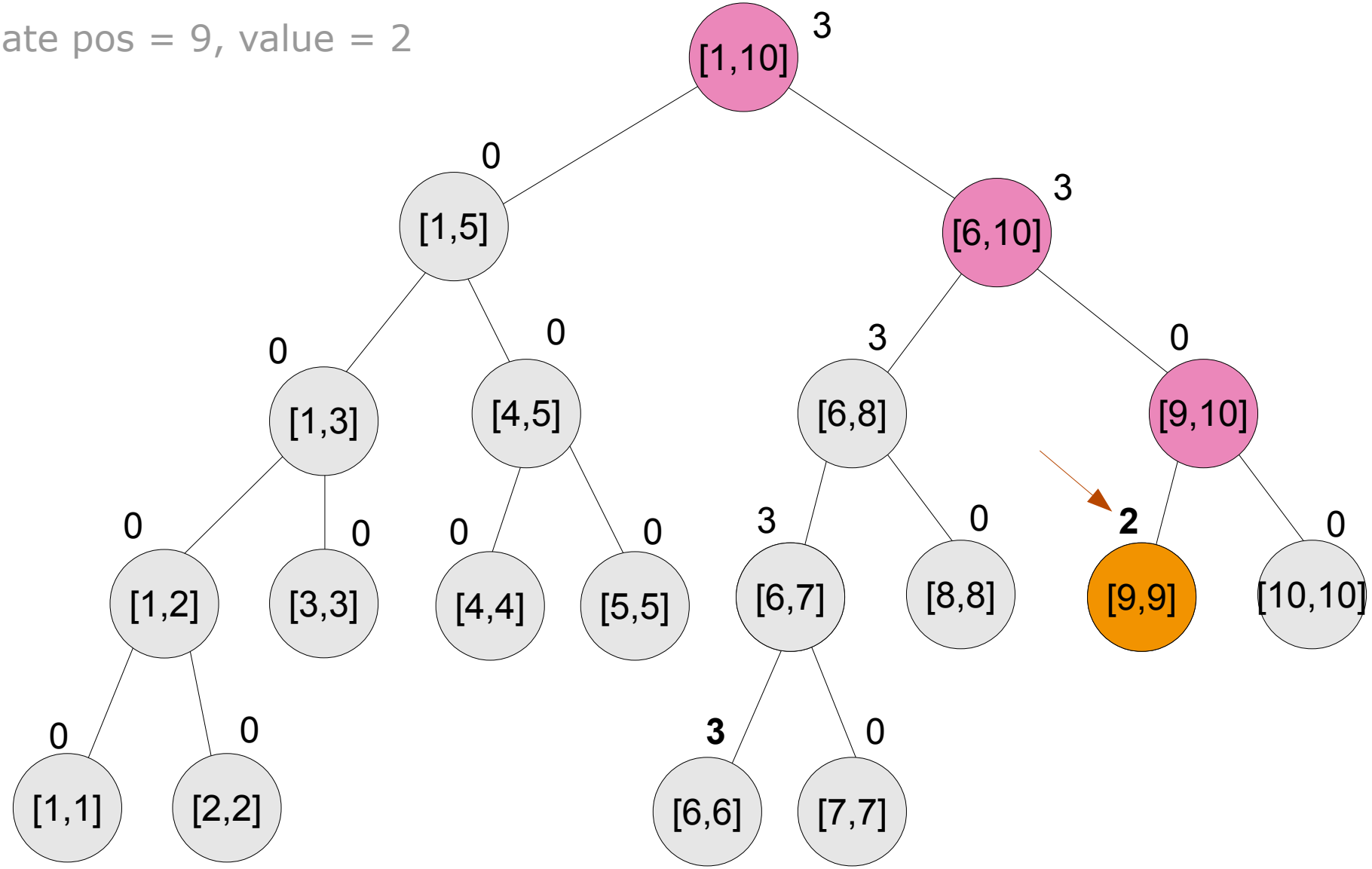
Interval Trees

Update pos = 9, value = 2



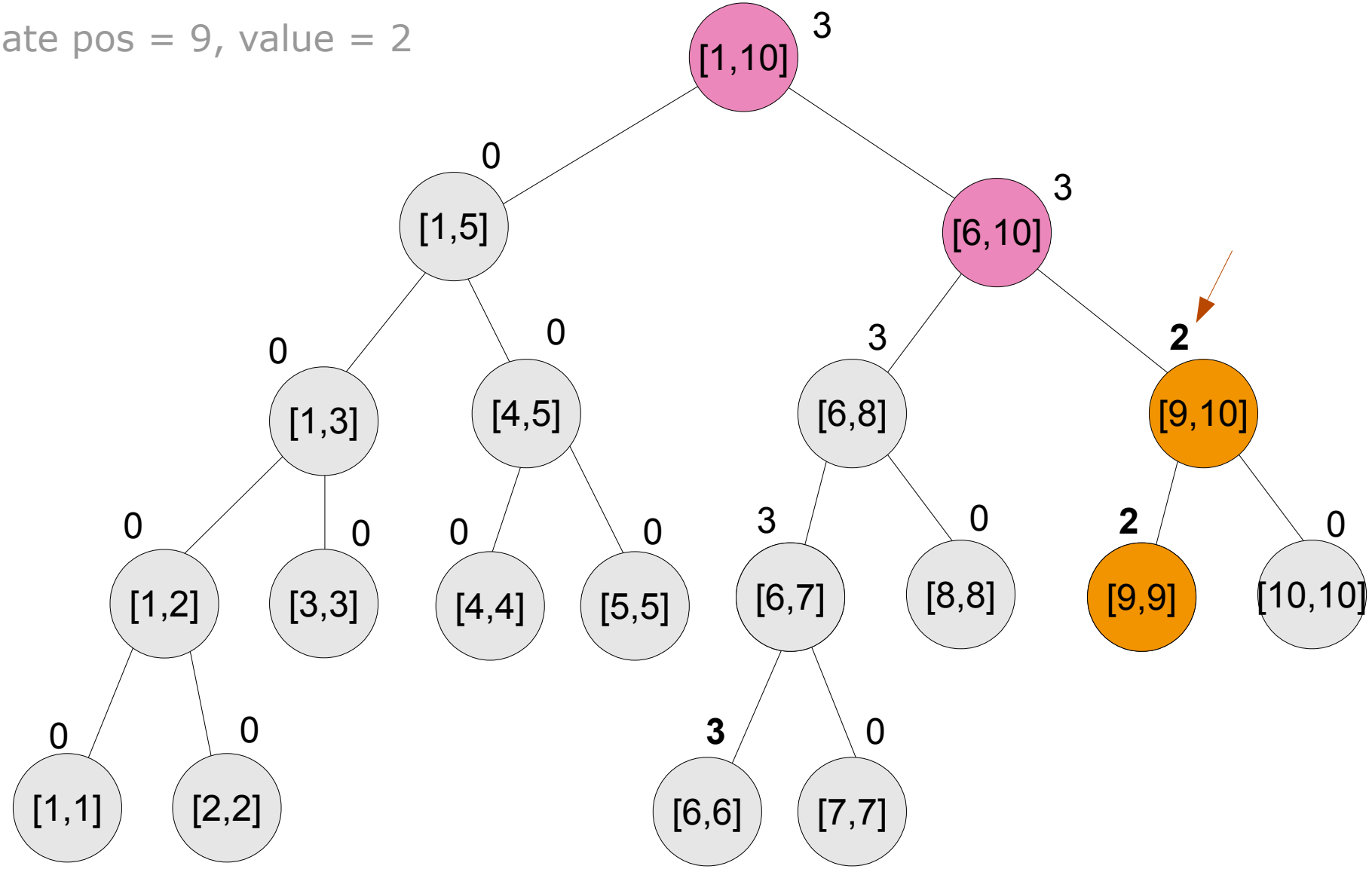
Interval Trees

Update pos = 9, value = 2



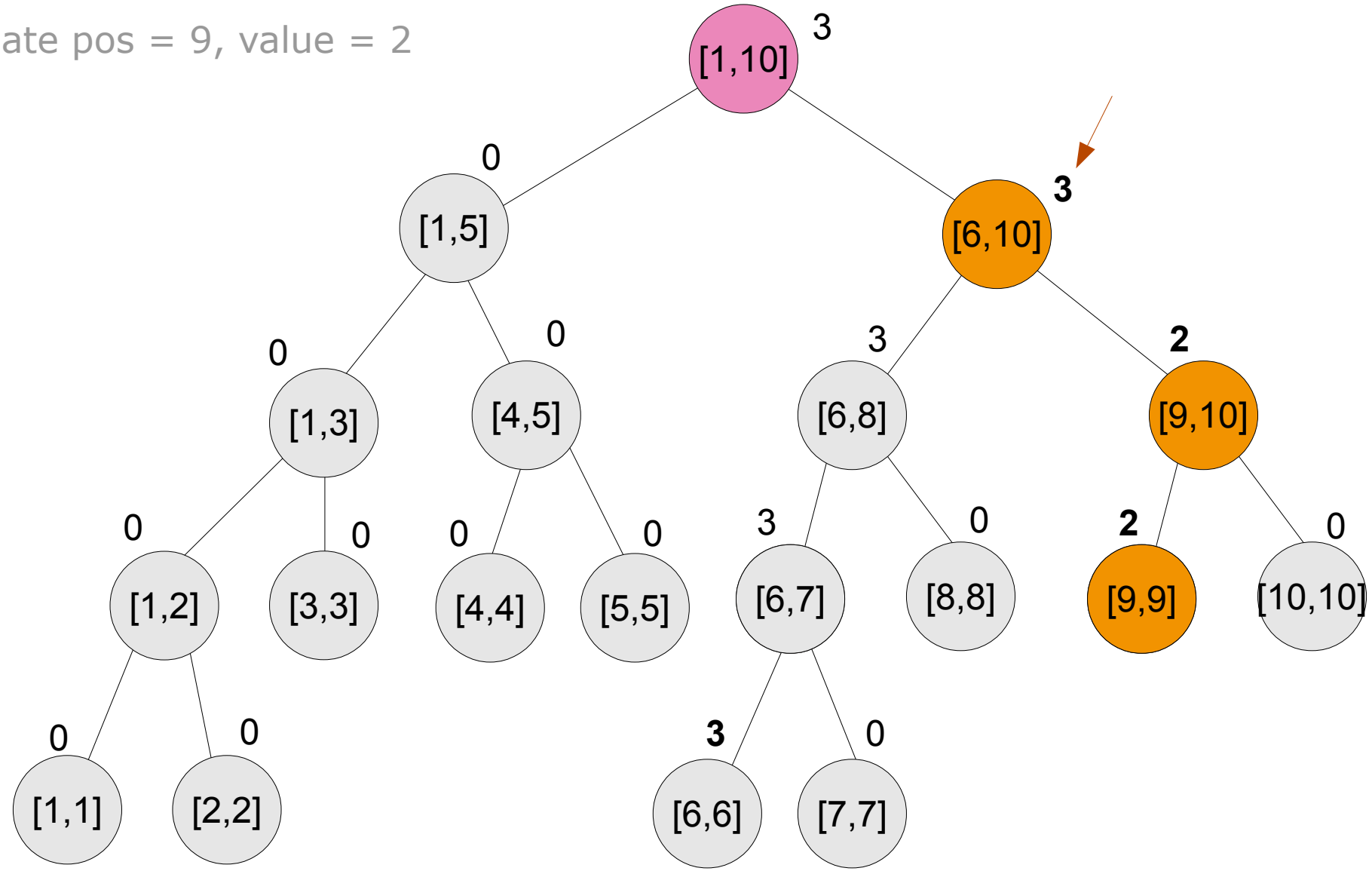
Interval Trees

Update pos = 9, value = 2



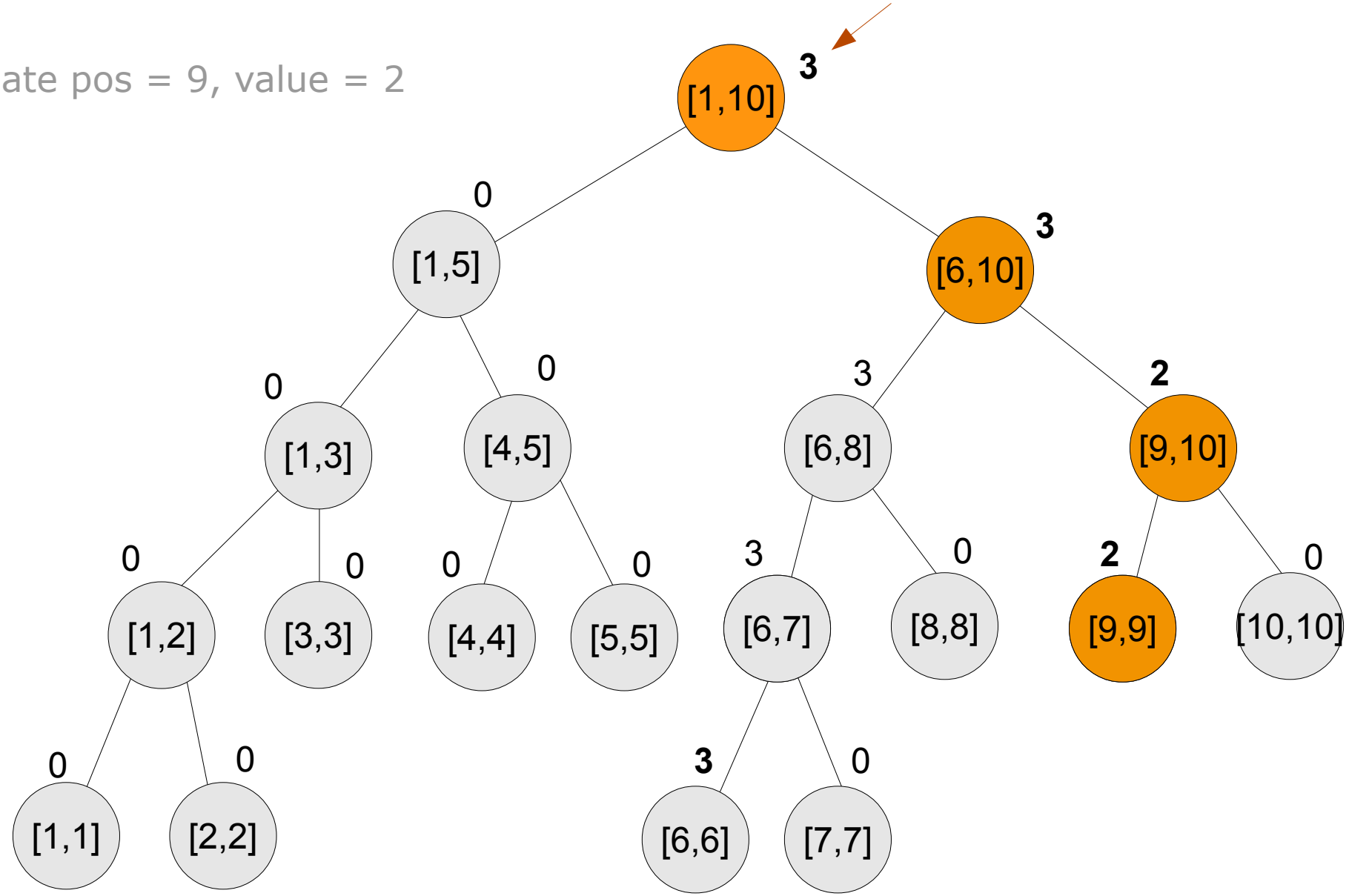
Interval Trees

Update pos = 9, value = 2



Interval Trees

Update pos = 9, value = 2



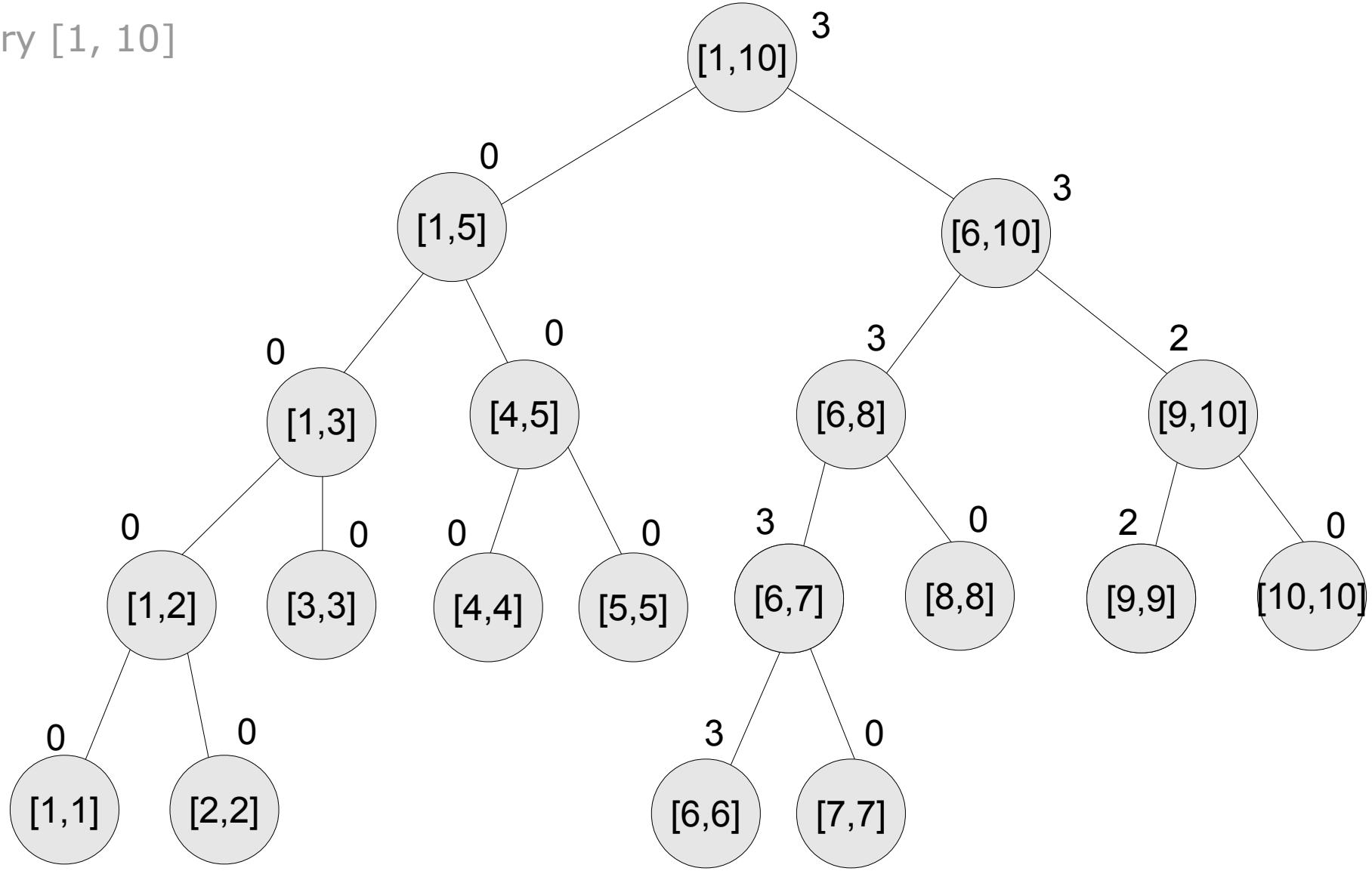
Interval Trees

```
int update(int pos, int value, int x, int y, int id) {
    int mid;
    if ( x == y ) {
        T[id] = value;
    }
    else {
        mid = (x + y)/2;
        if ( pos <= mid ) {
            update(pos, value, x, mid, 2*id);
        }
        else {
            update(pos, value, mid+1, y, 2*id + 1);
        }
        T[id] = max(T[2*id], T[2*id + 1]);
    }
}


update(pos, value, 1, N, 1);
```

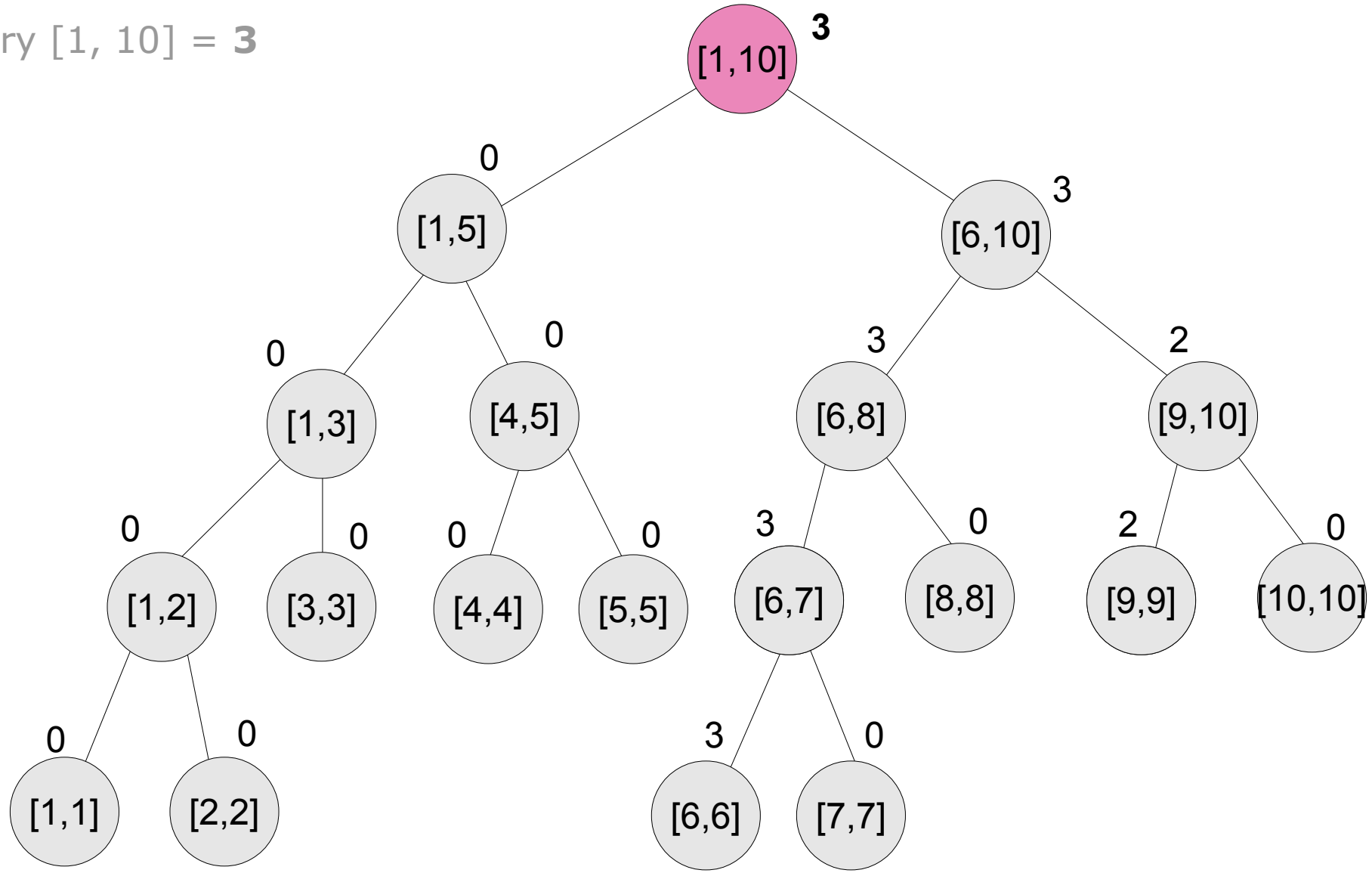
Interval Trees

Query [1, 10]



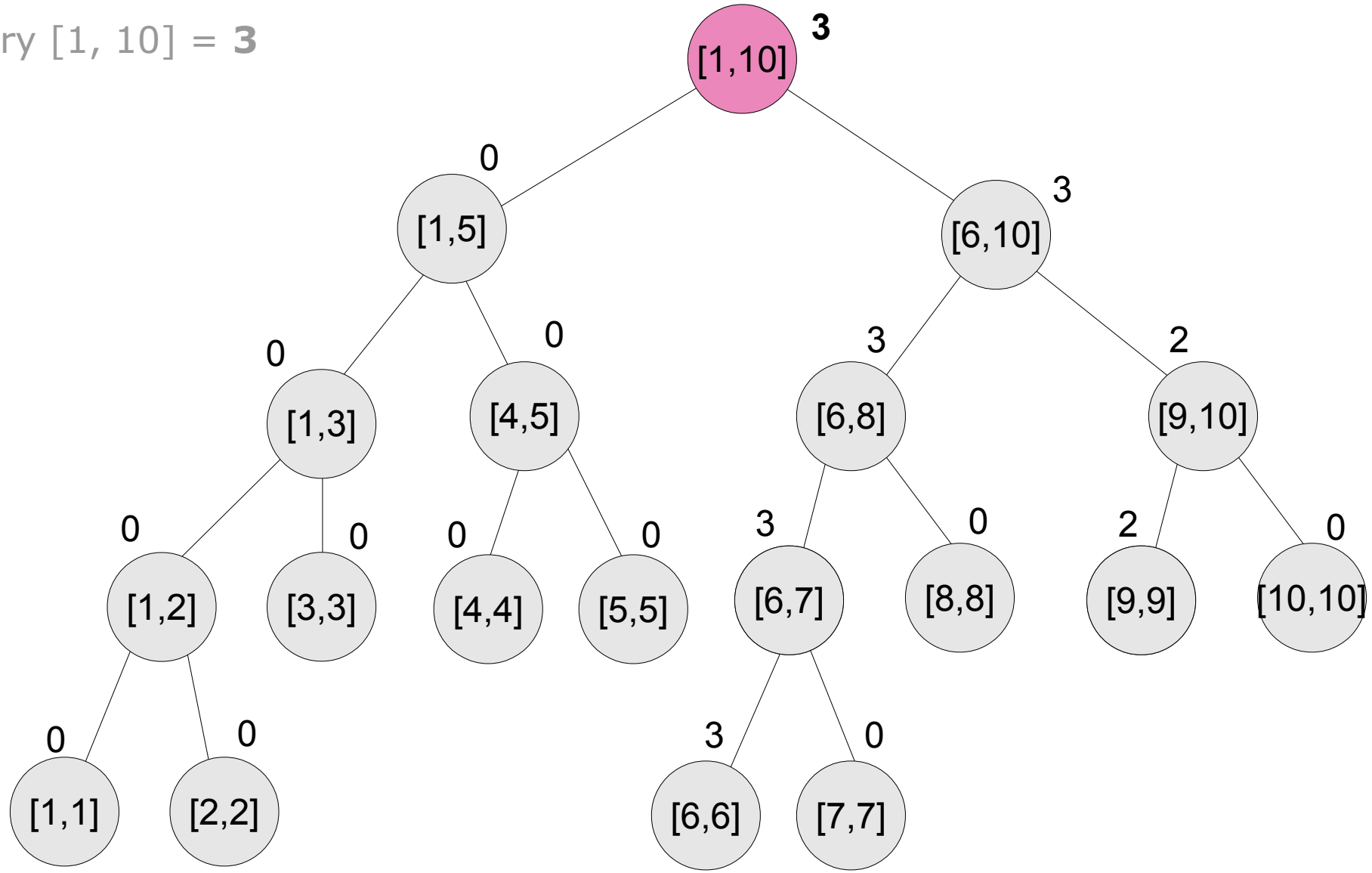
Interval Trees

Query [1, 10] = **3**



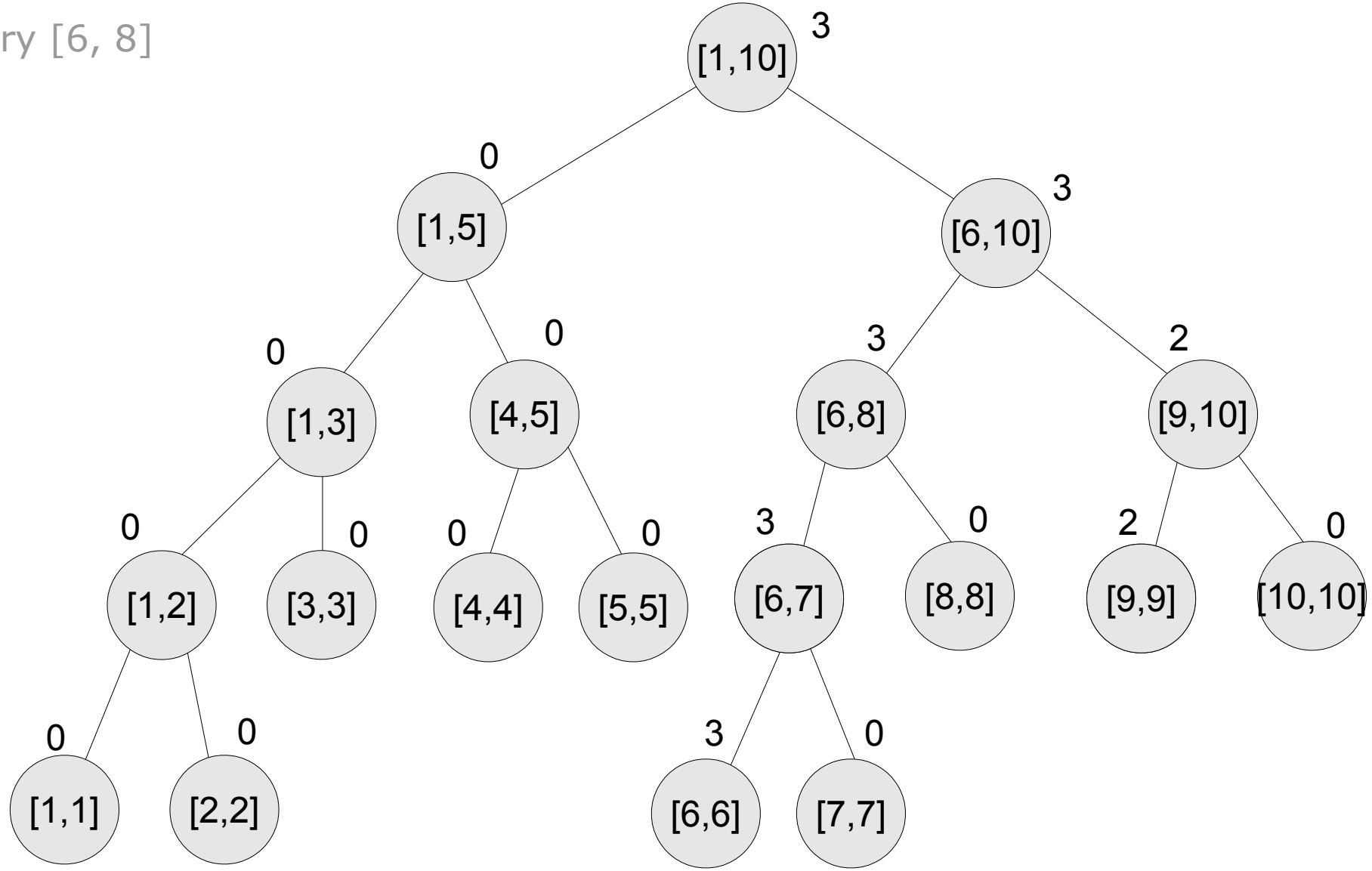
Interval Trees

Query [1, 10] = **3**



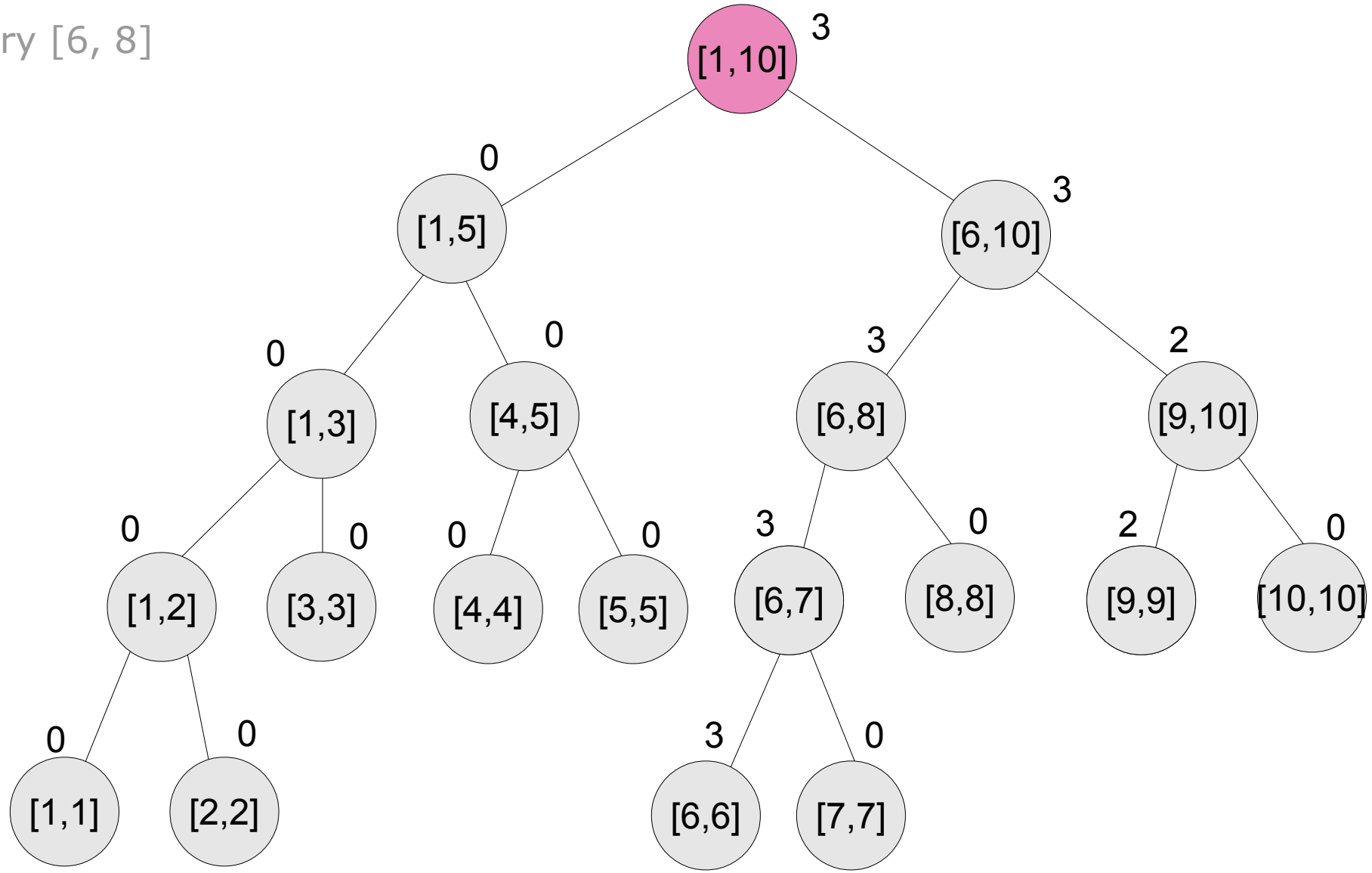
Interval Trees

Query [6, 8]



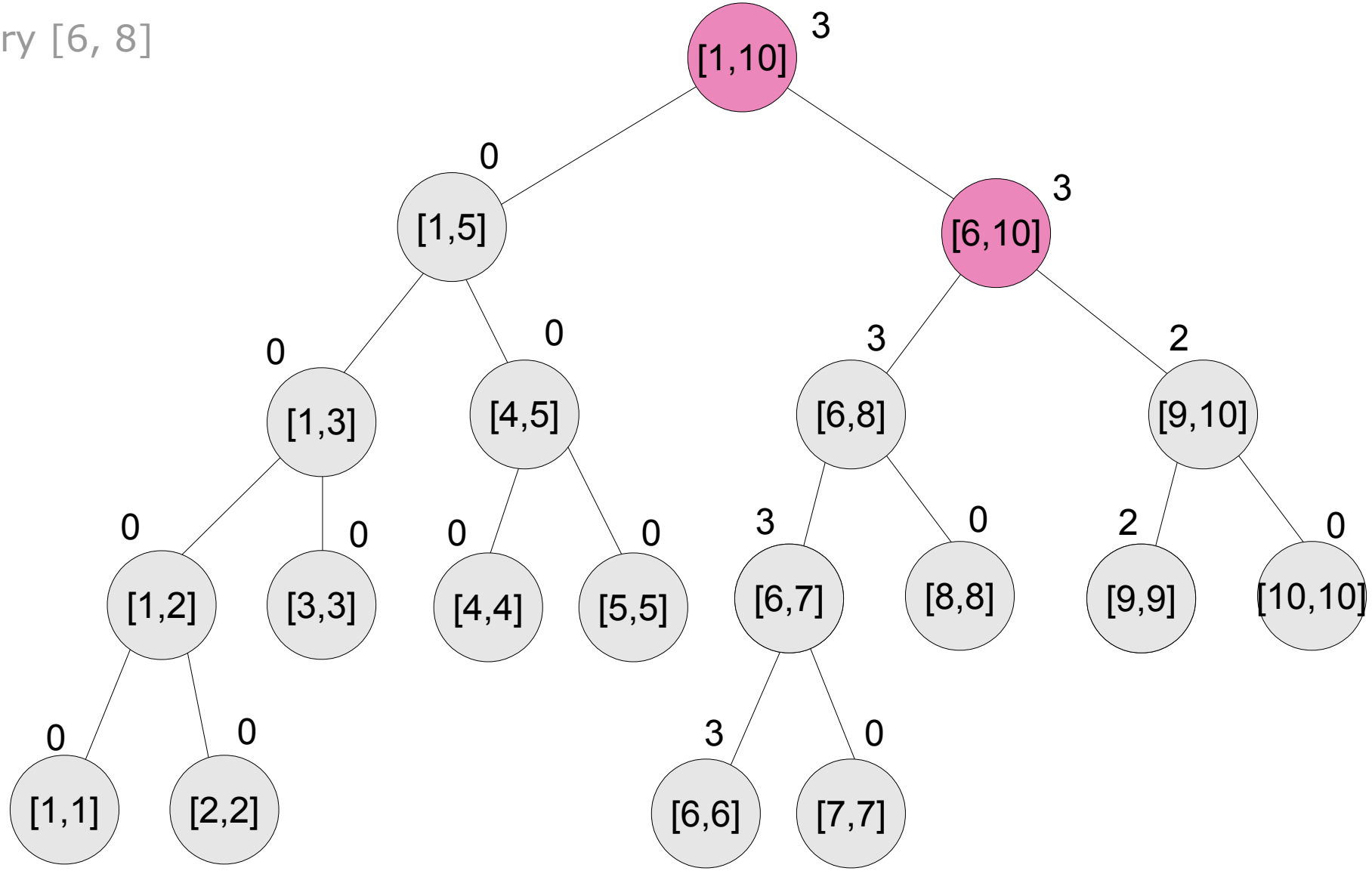
Interval Trees

Query [6, 8]



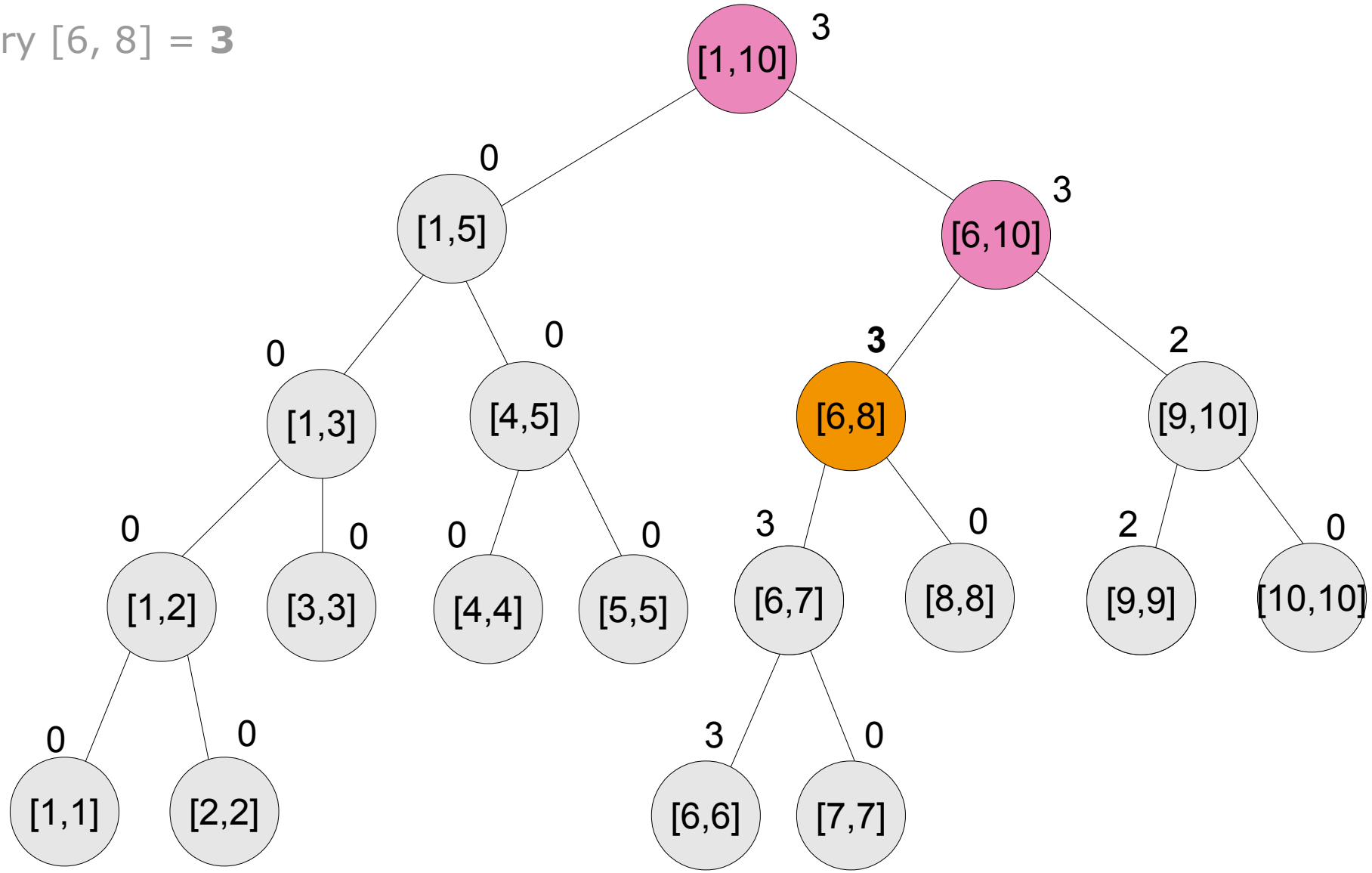
Interval Trees

Query [6, 8]



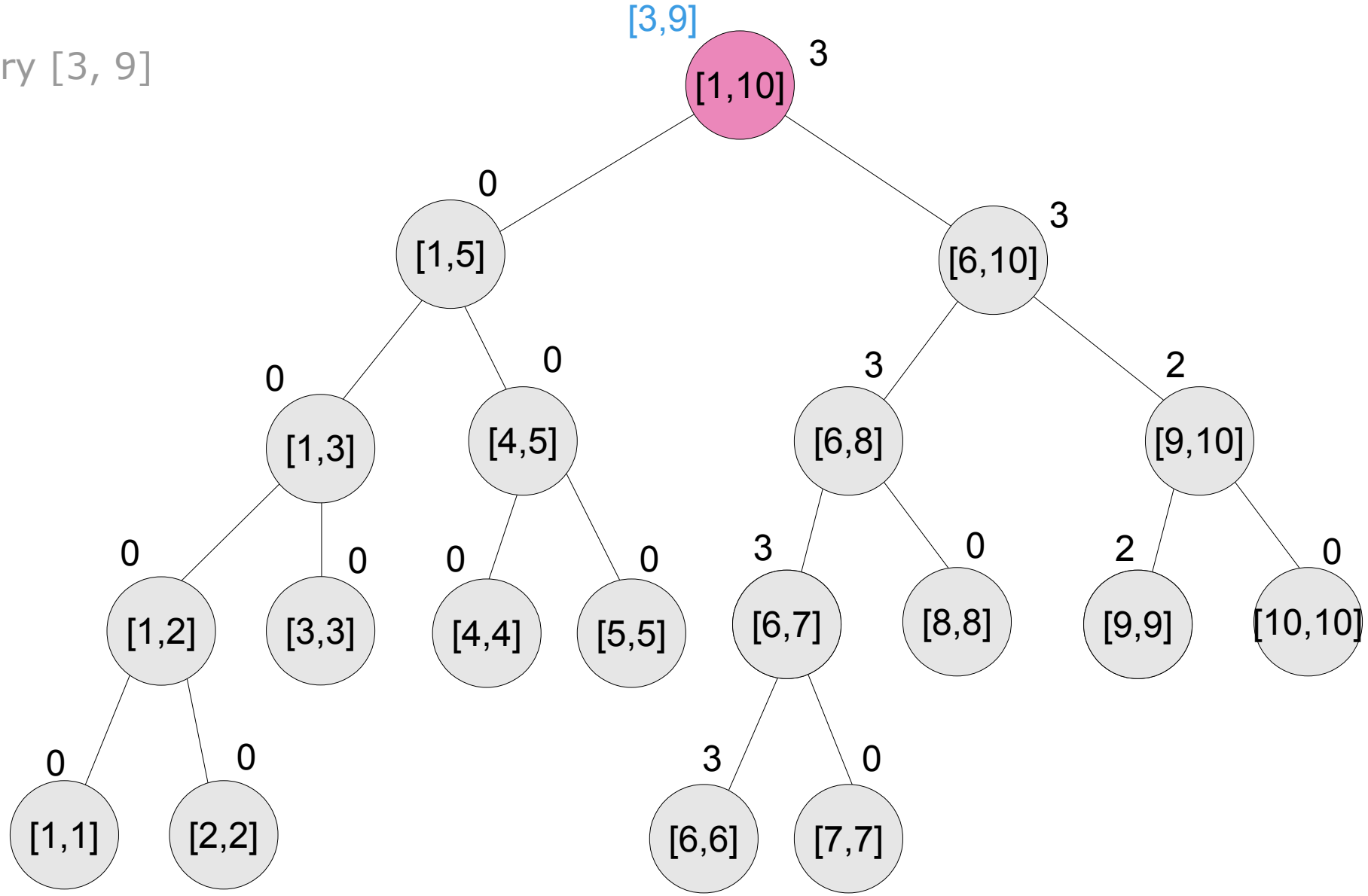
Interval Trees

Query [6, 8] = **3**



Interval Trees

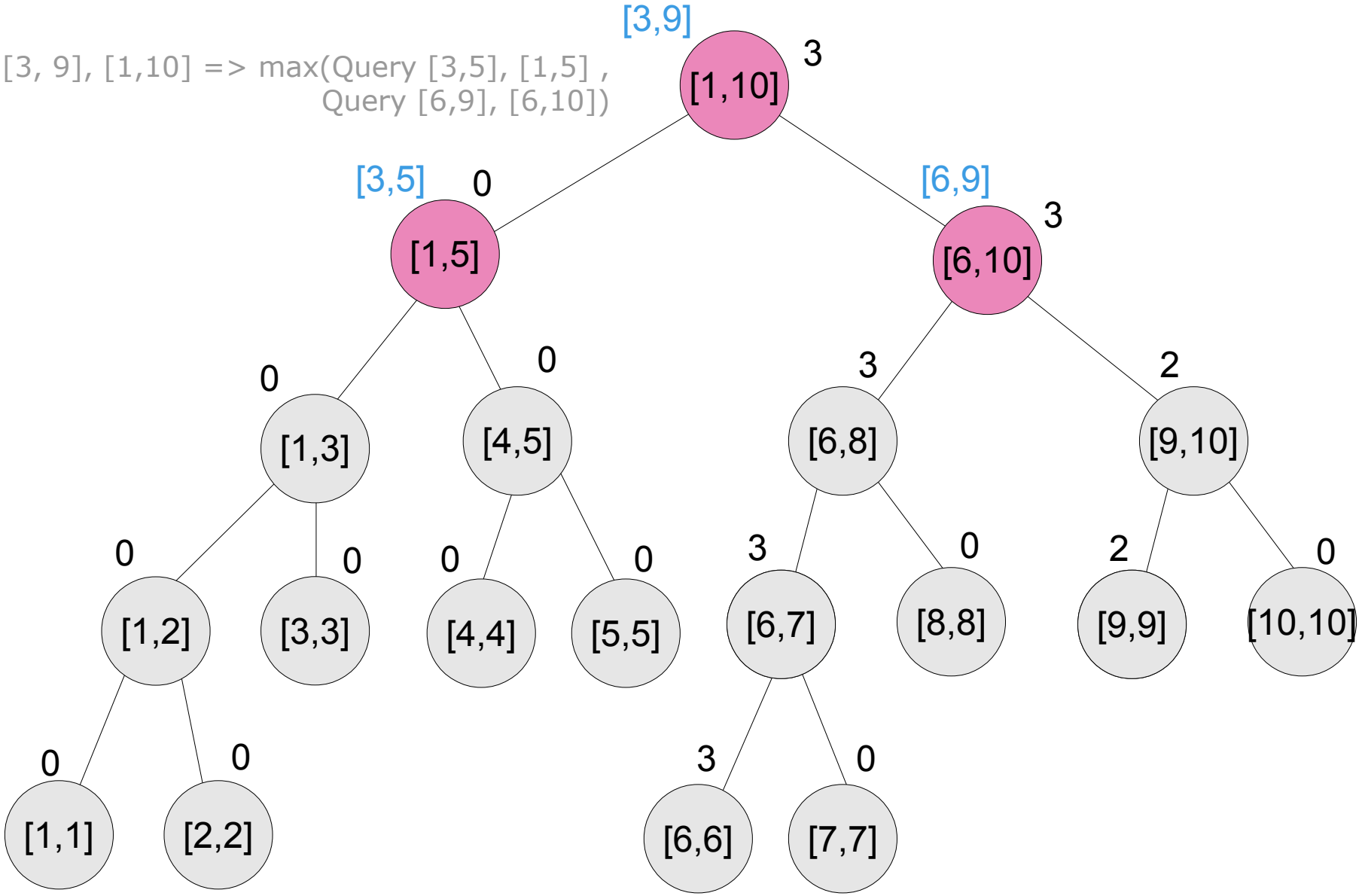
Query [3, 9]



Query [3, 9]

Interval Trees

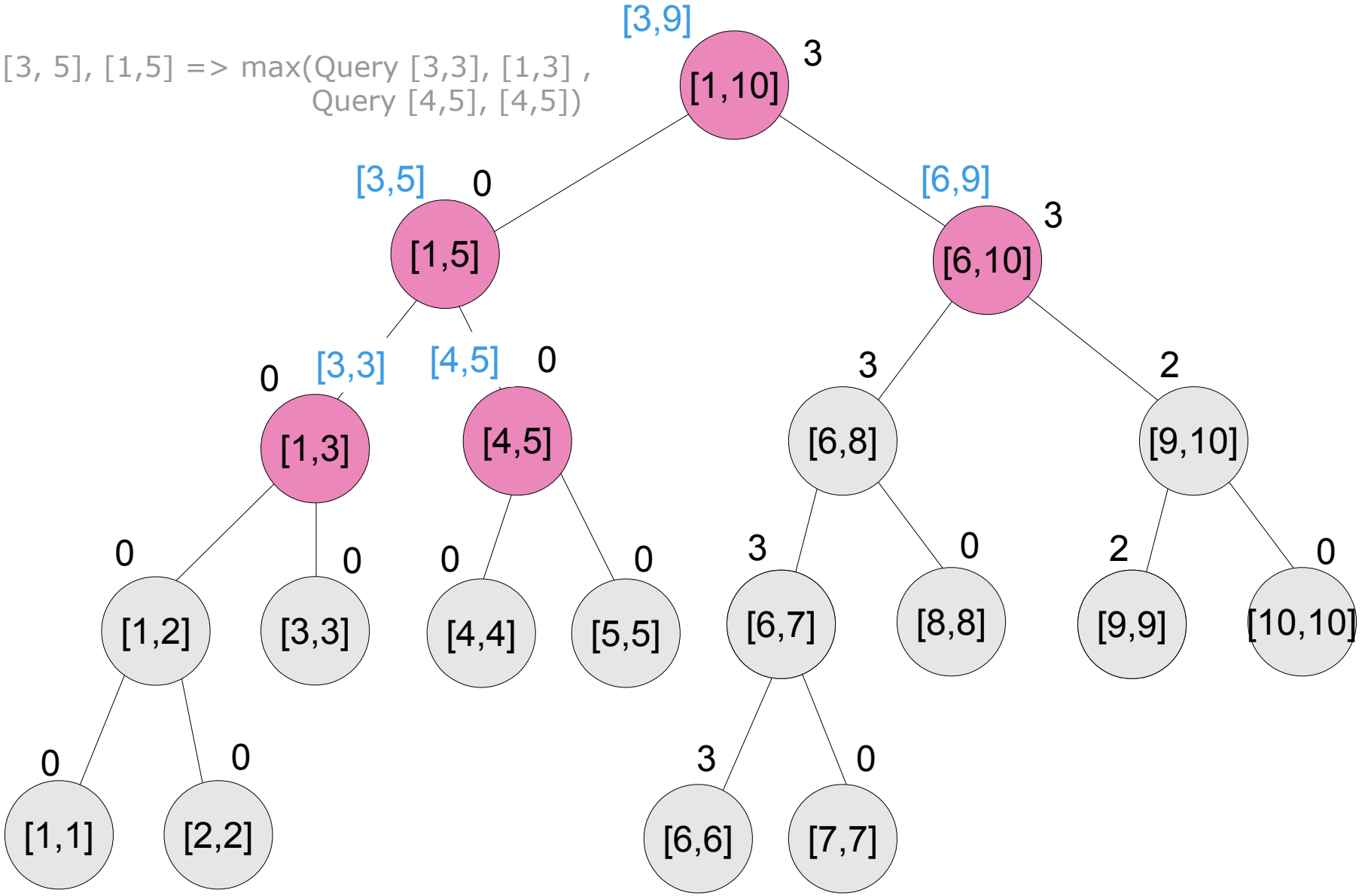
Query [3, 9], [1,10] => max(Query [3,5], [1,5] ,
Query [6,9], [6,10])



Interval Trees

Query [3, 9]

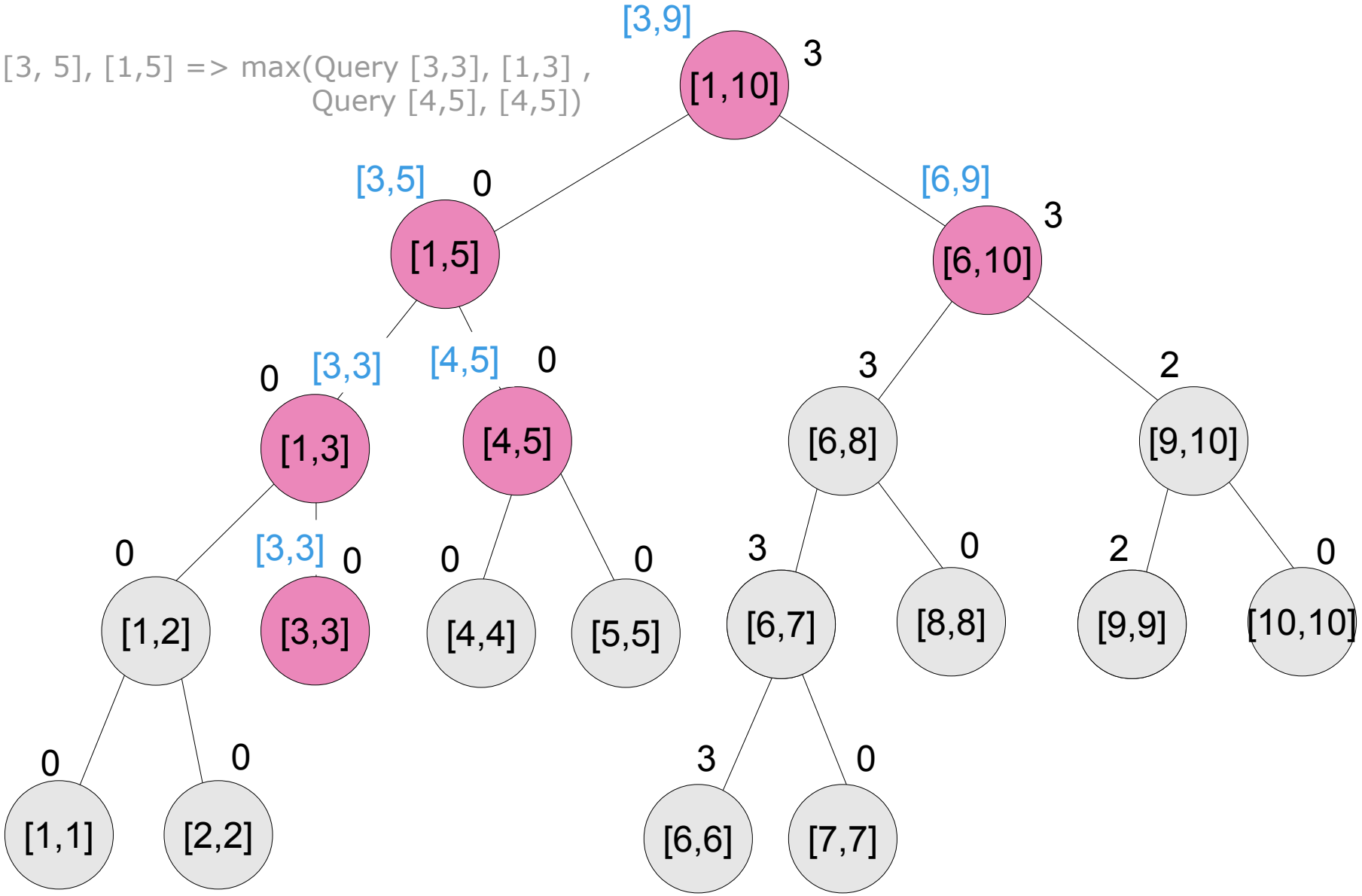
Query [3, 5], [1,5] => max(Query [3,3], [1,3] , Query [4,5], [4,5])



Interval Trees

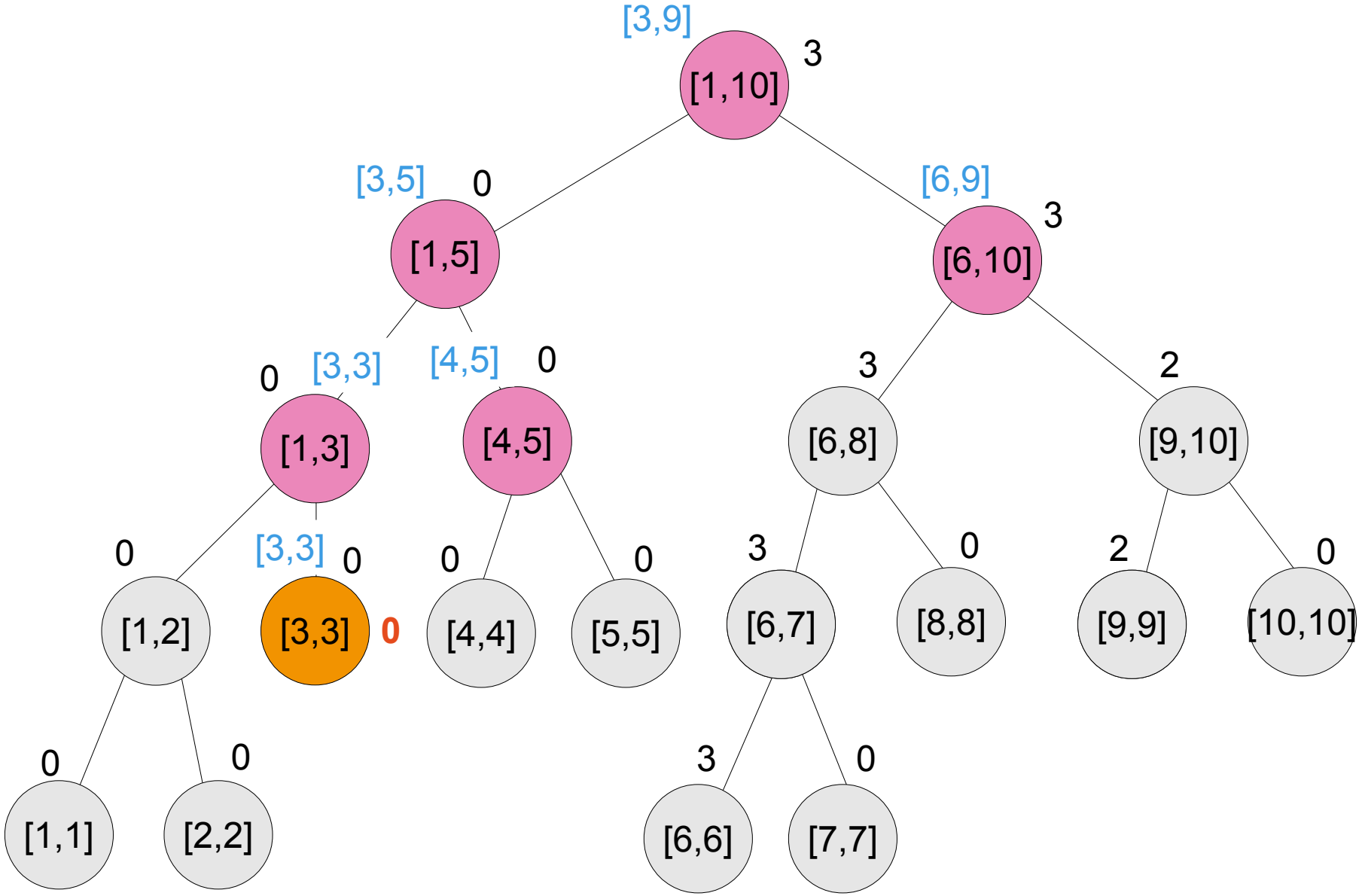
Query [3, 9]

Query [3, 5], [1,5] => max(Query [3,3], [1,3] ,
Query [4,5], [4,5])



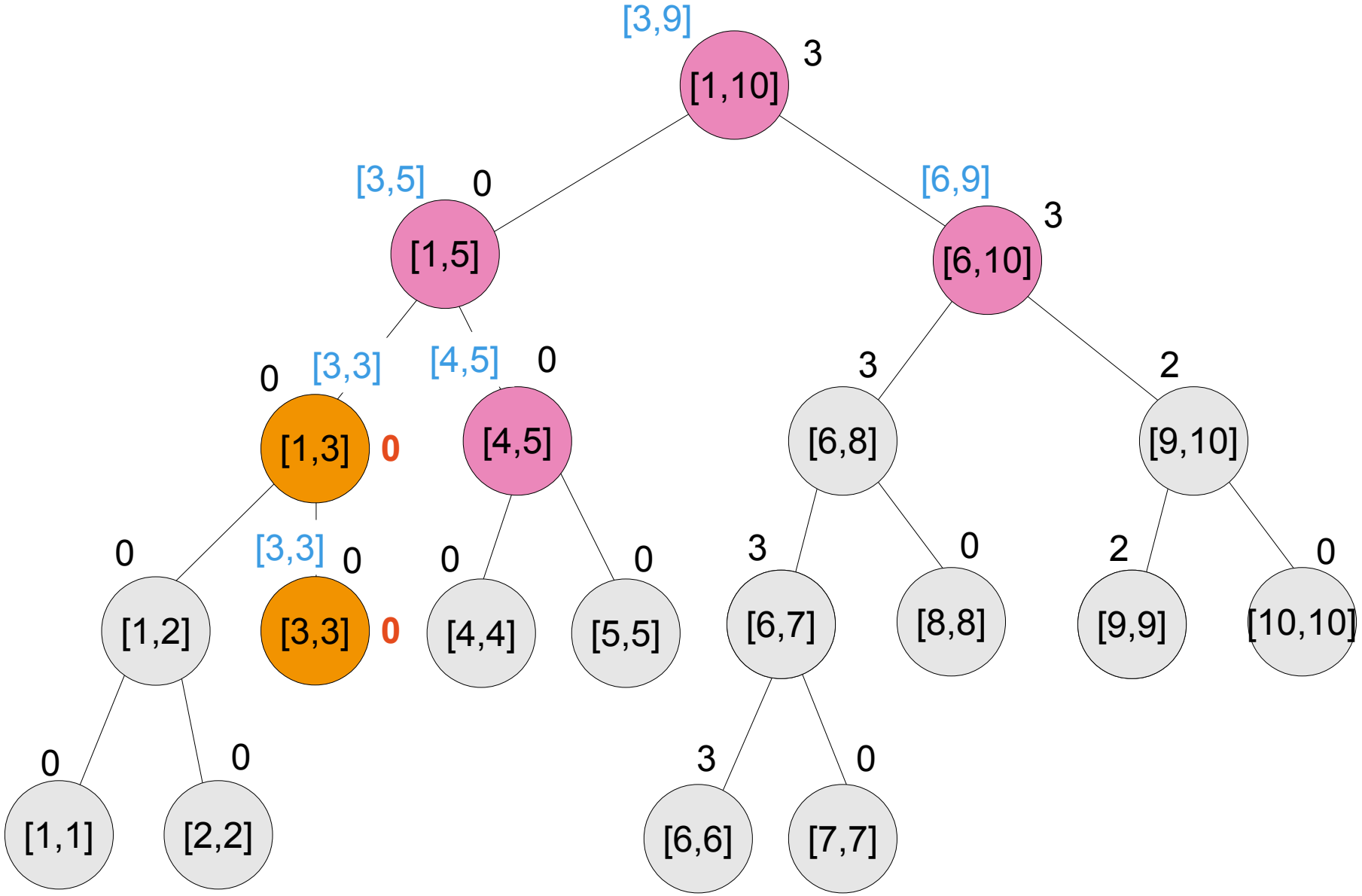
Query [3, 9]

Interval Trees



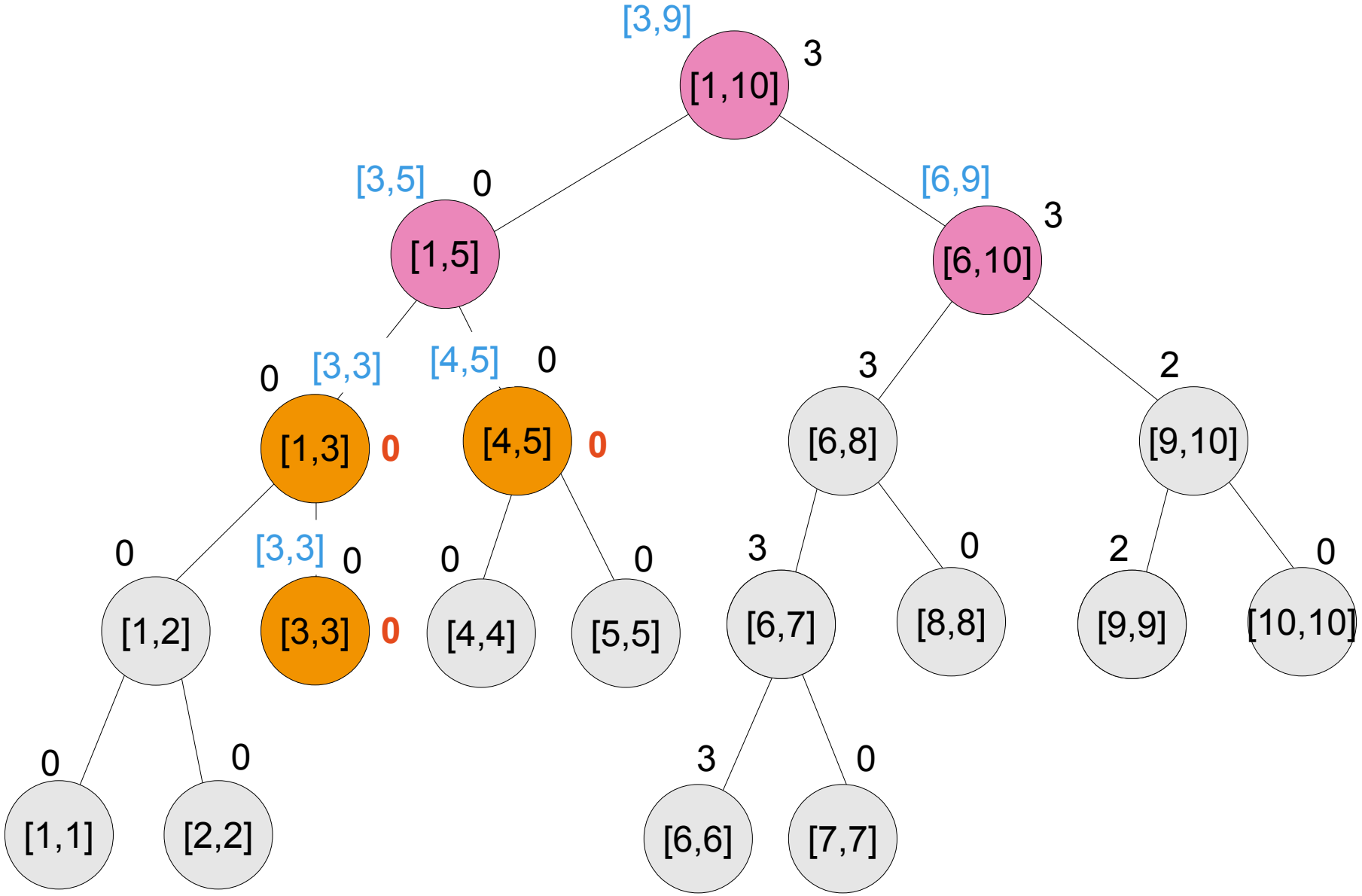
Query [3, 9]

Interval Trees



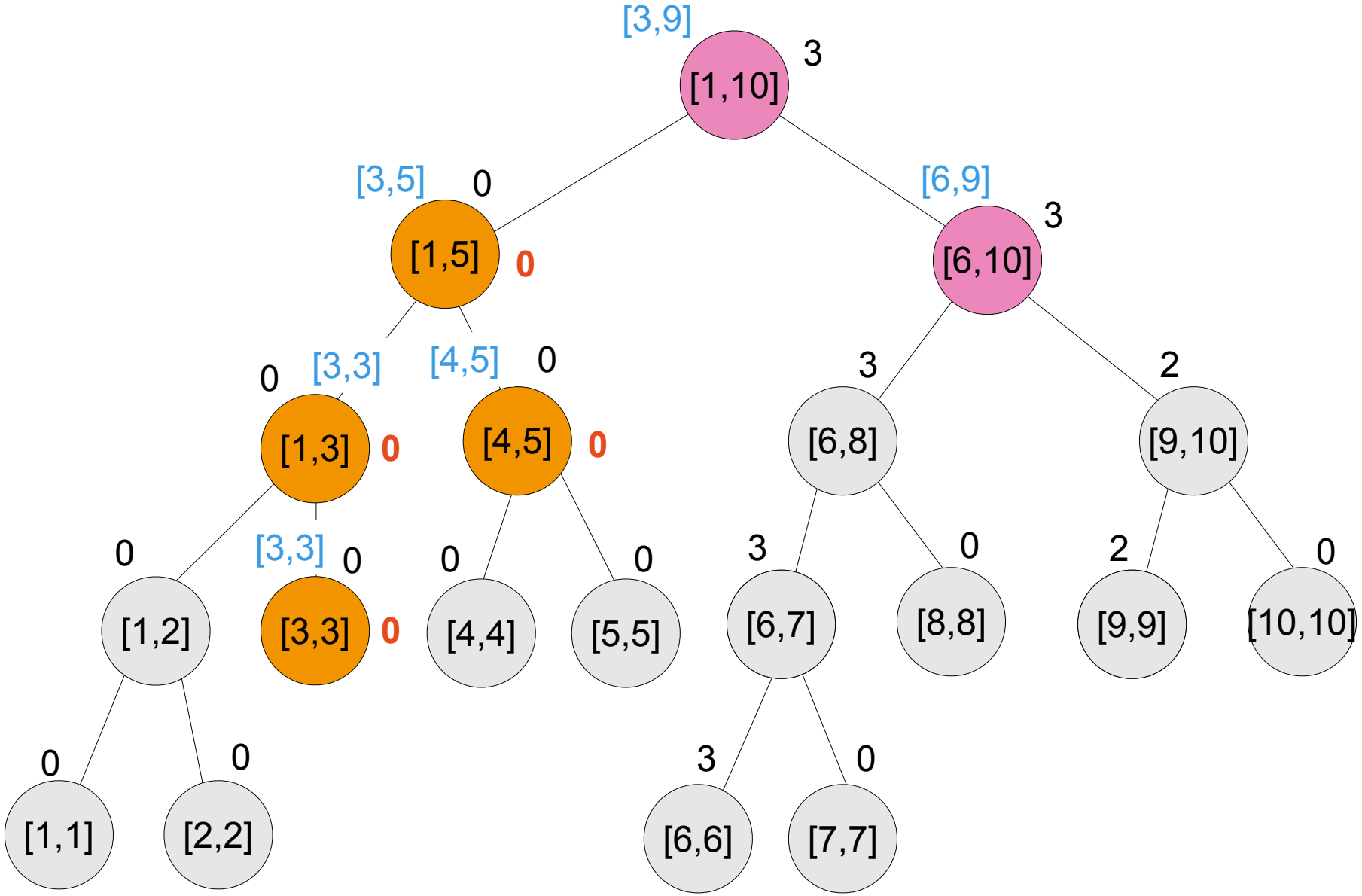
Query [3, 9]

Interval Trees



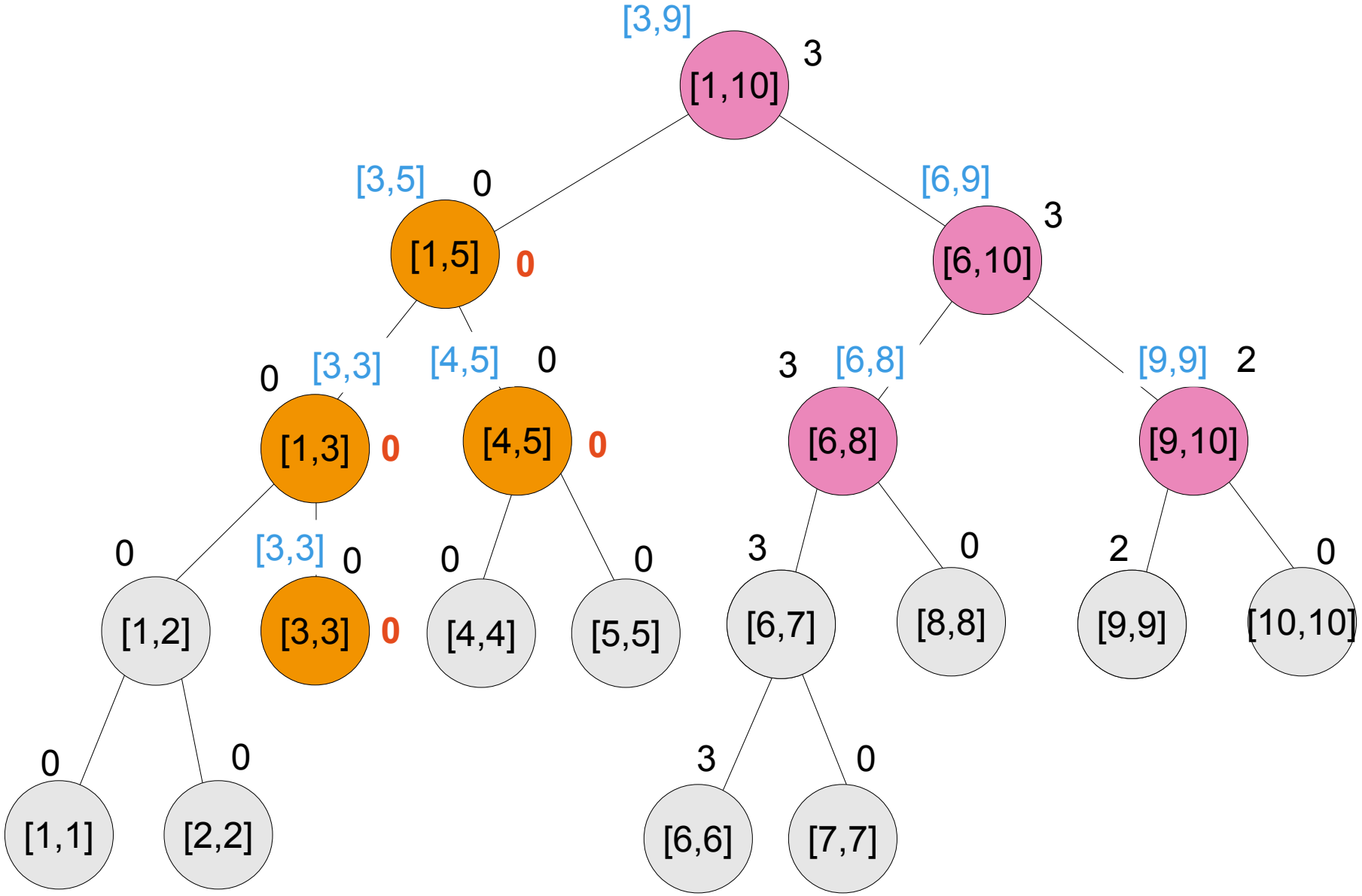
Query [3, 9]

Interval Trees



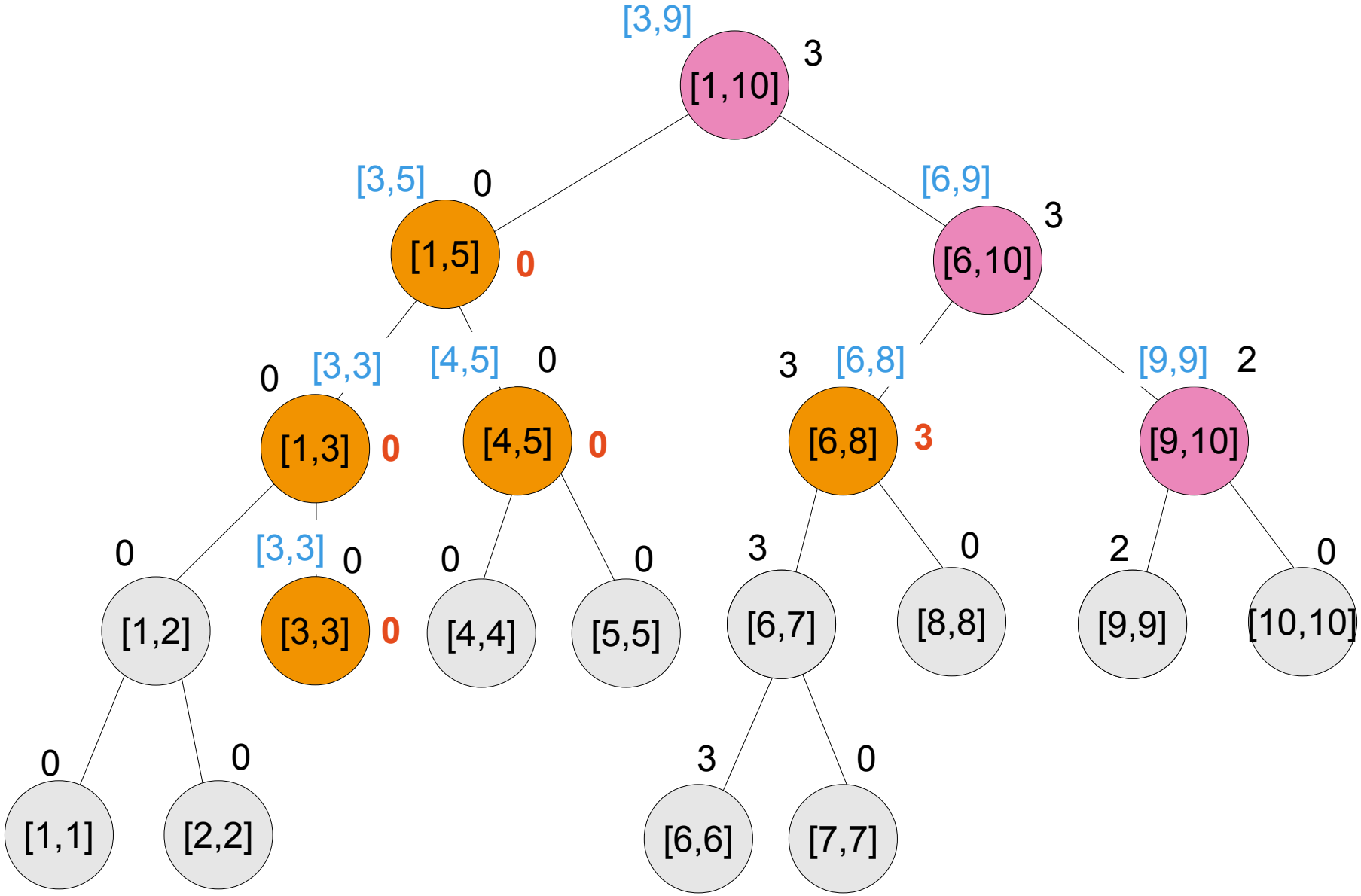
Query [3, 9]

Interval Trees



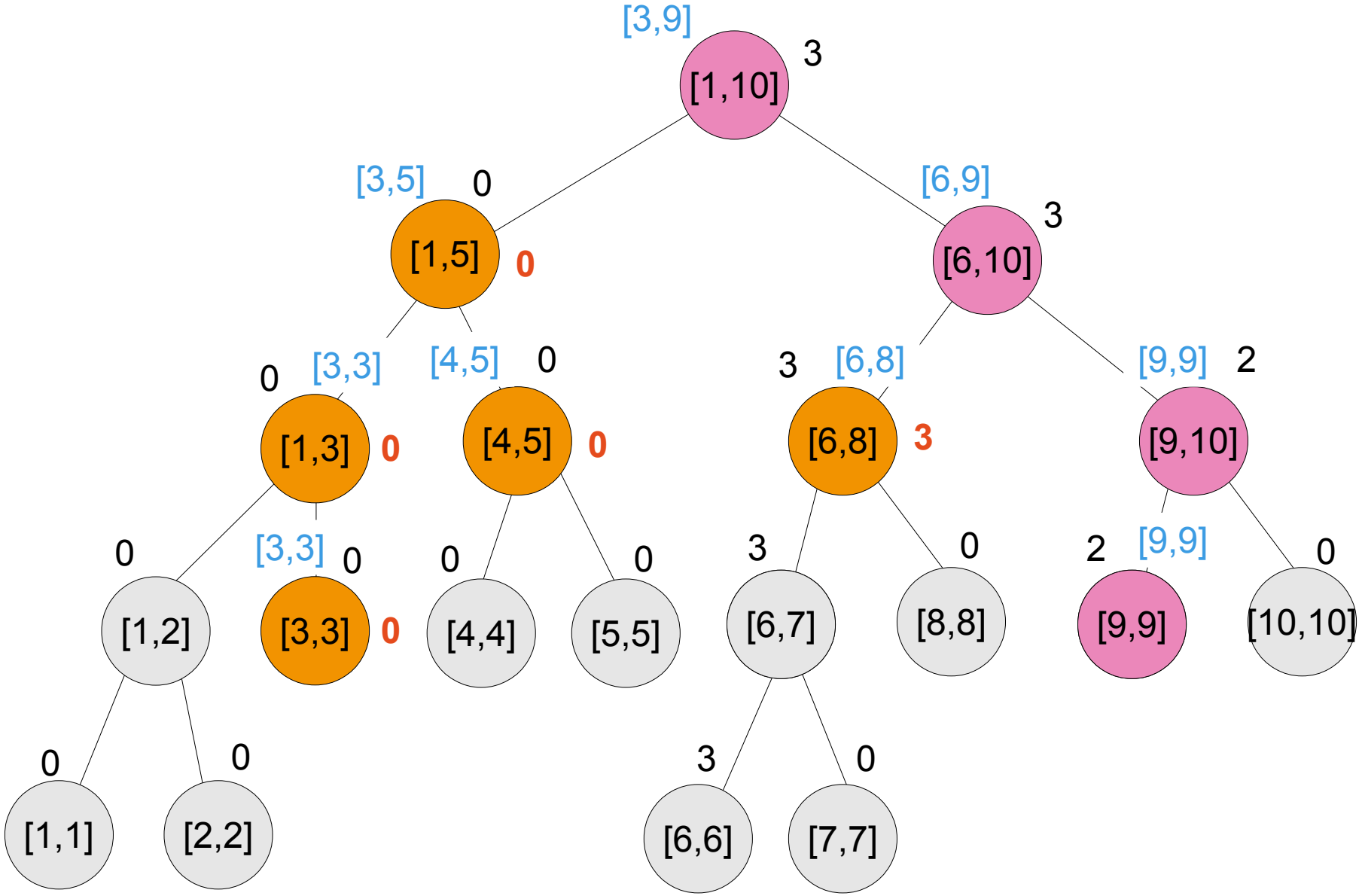
Query [3, 9]

Interval Trees



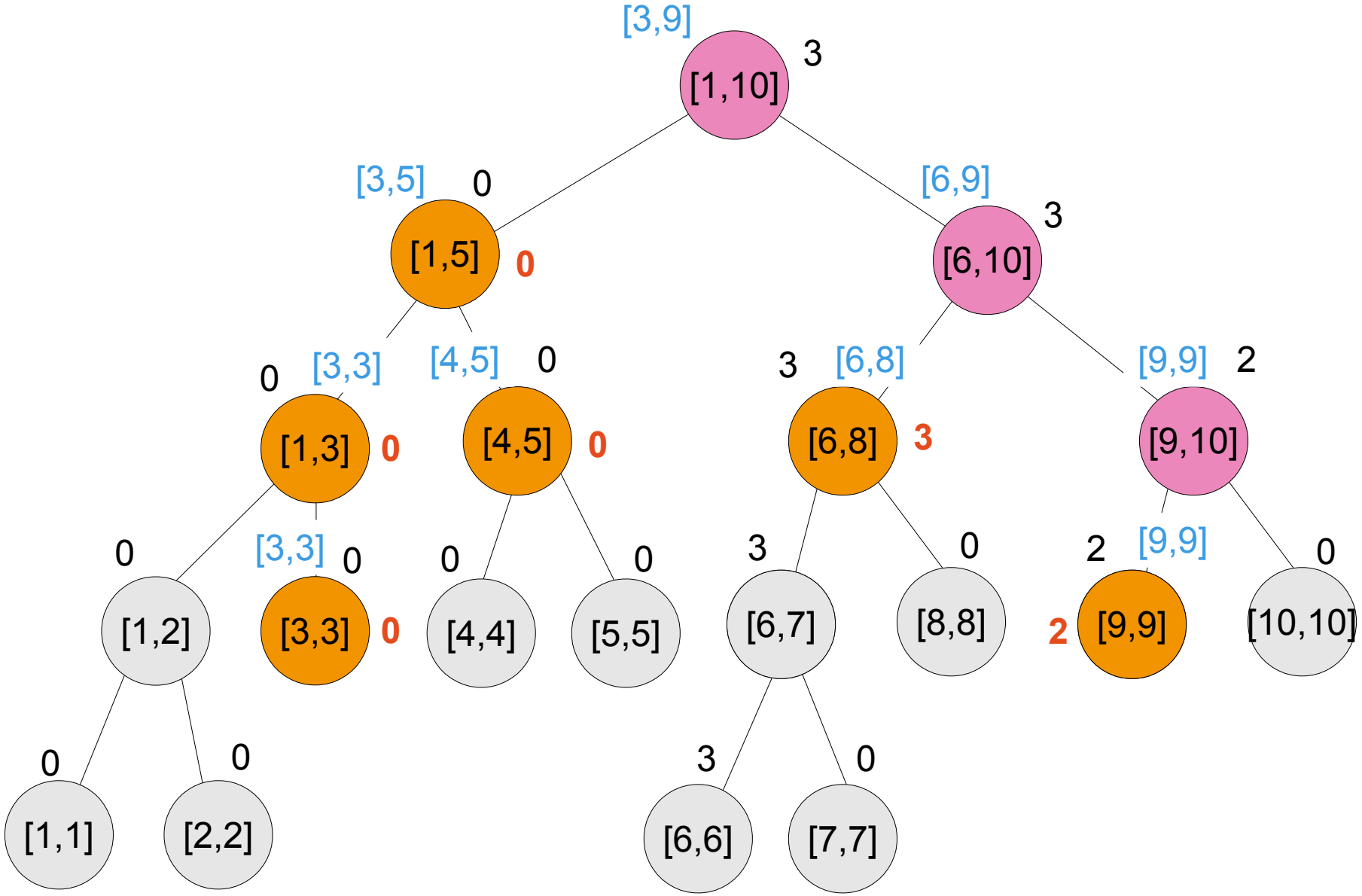
Query [3, 9]

Interval Trees



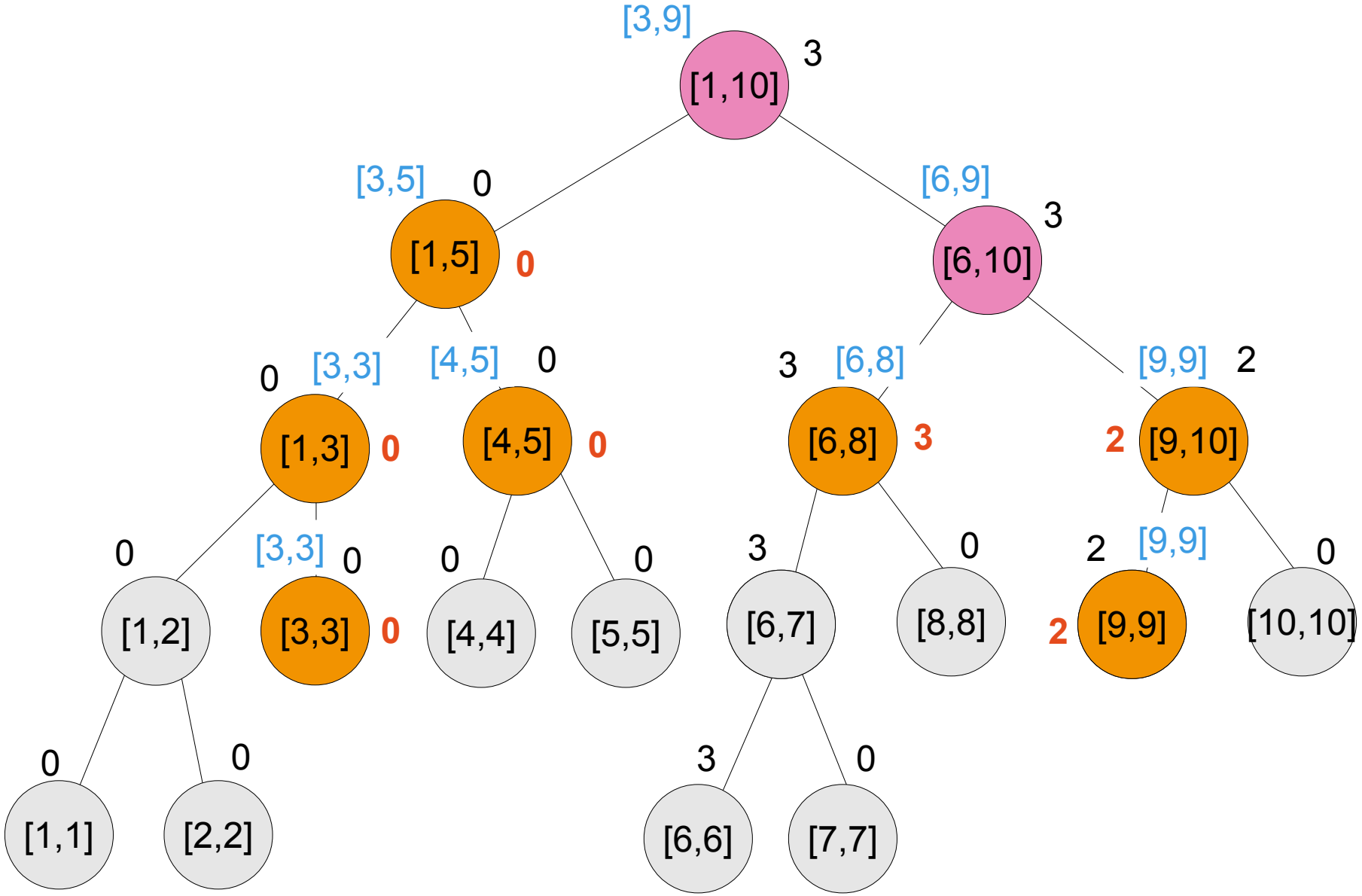
Query [3, 9]

Interval Trees



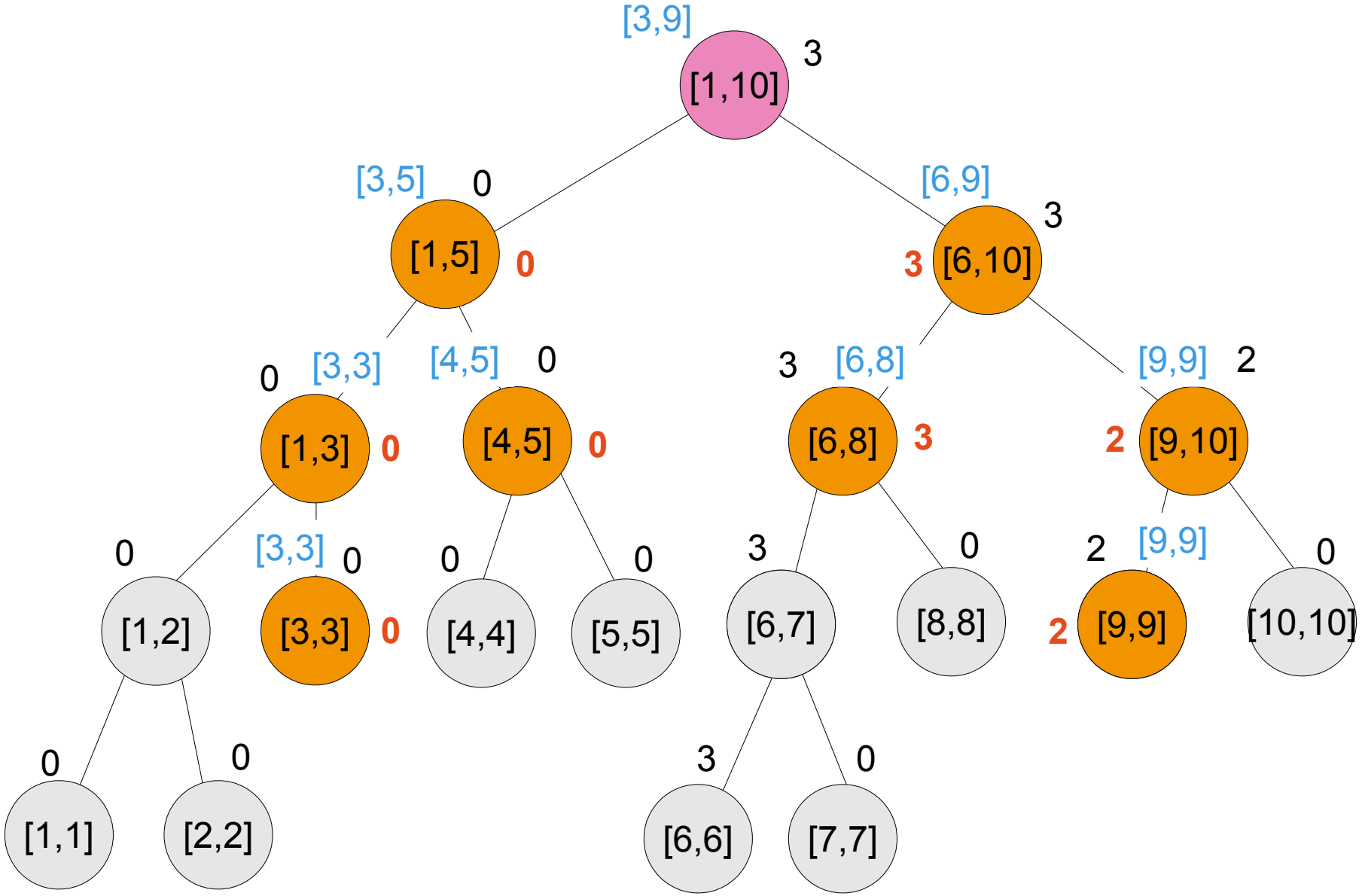
Query [3, 9]

Interval Trees



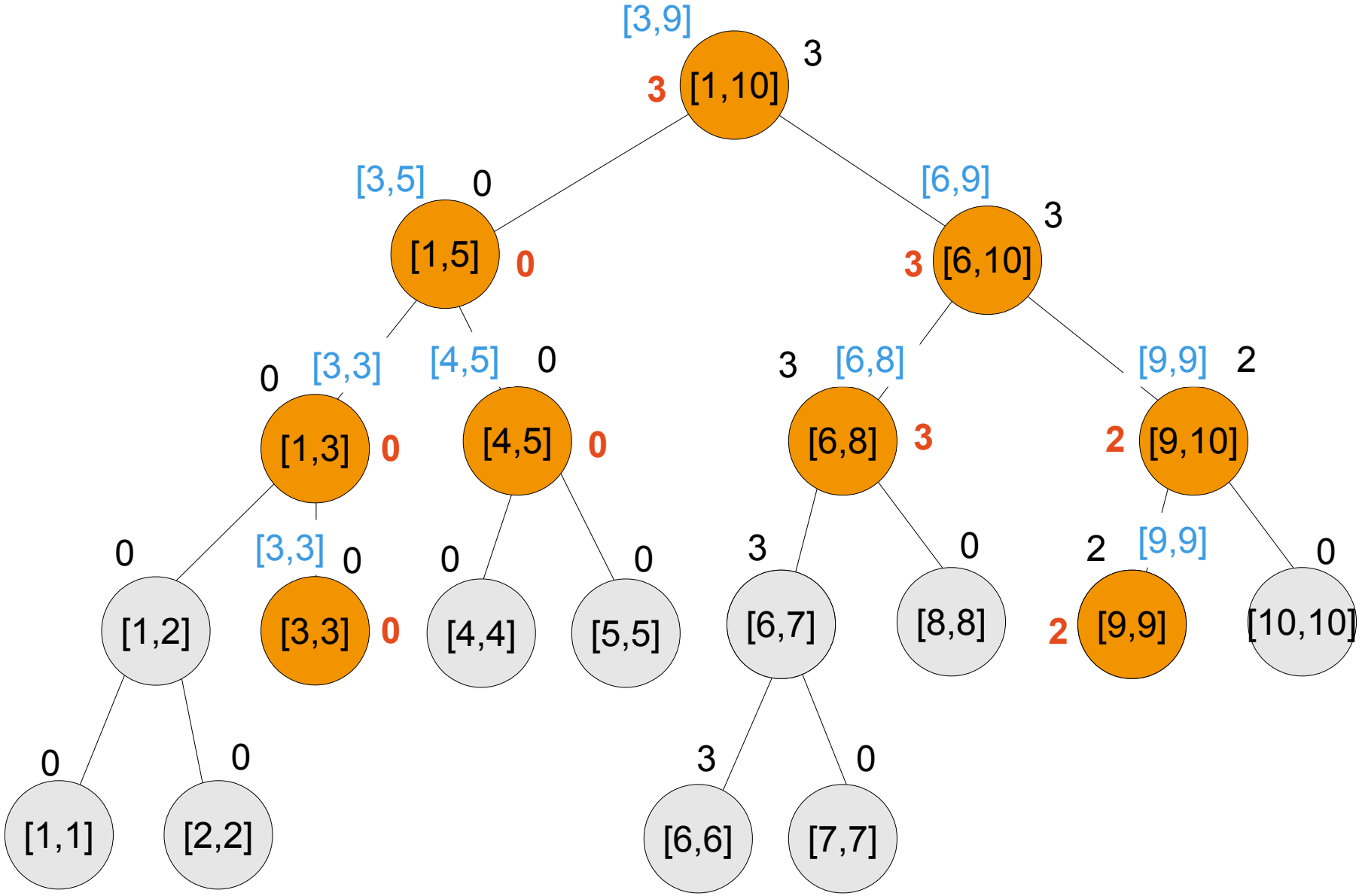
Query [3, 9]

Interval Trees

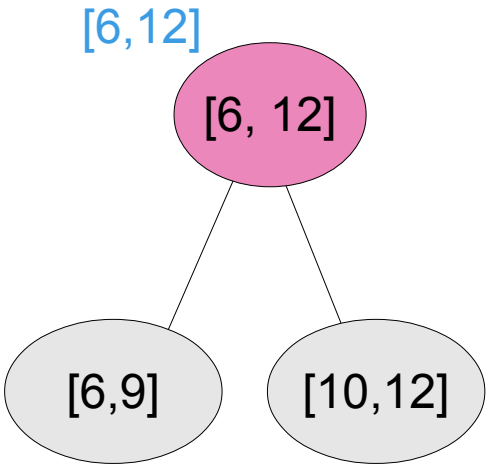


Query [3, 9]

Interval Trees

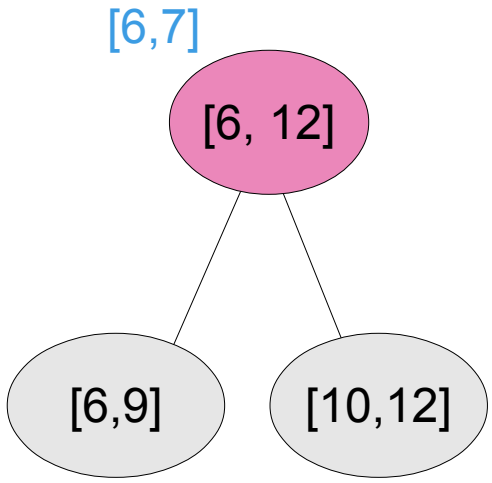


Περιπτώσεις ερωτημάτων (1η)



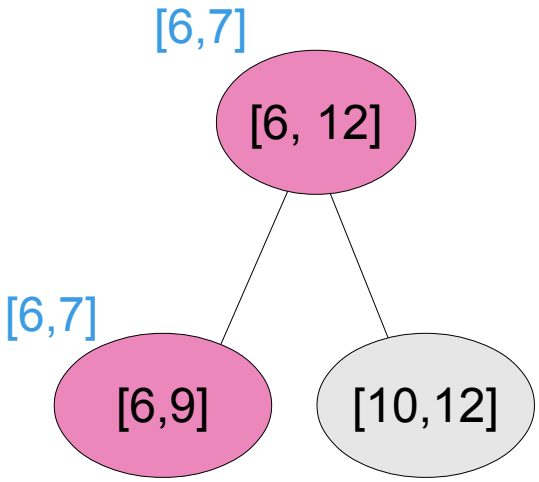
QX = x και QY = y
επιστρέφουμε T[id]

Περιπτώσεις ερωτημάτων (2η)



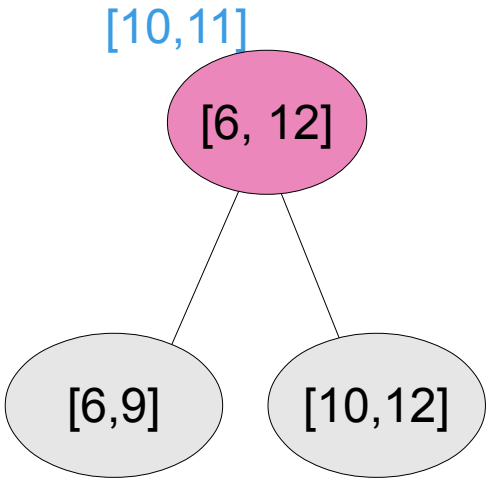
QY <= mid

μας ενδιαφέρει μόνο το αριστερό κλαδί



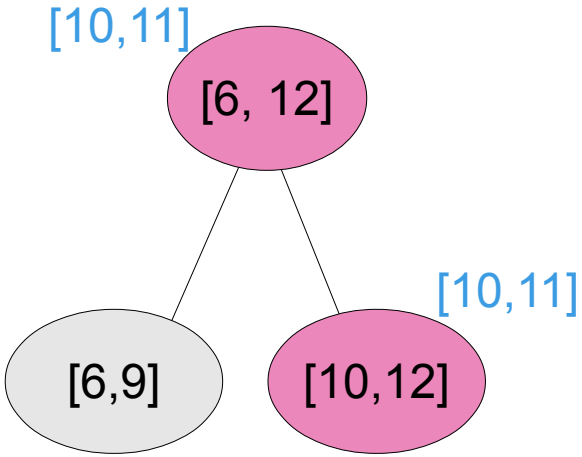
Επιστρέφουμε το αποτέλεσμα του αριστερού κλαδιού στο query με το ίδιο διάστημα

Περιπτώσεις ερωτημάτων (3η)



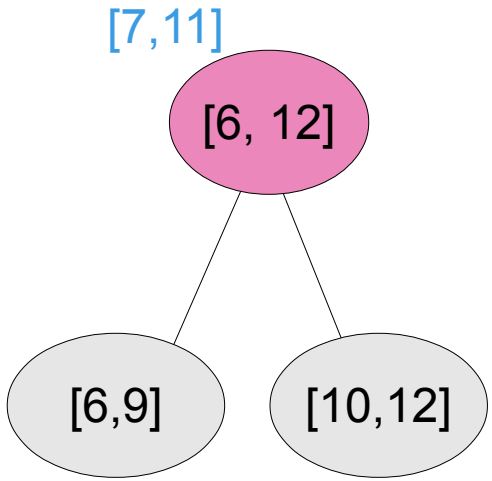
QX >= mid + 1

μας ενδιαφέρει μόνο το δεξιό κλαδί

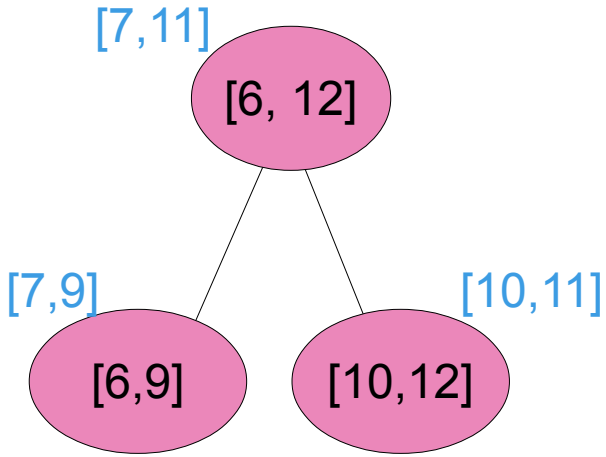


Επιστρέφουμε το αποτέλεσμα του δεξιού κλαδιού στο query με το ίδιο διάστημα

Περιπτώσεις ερωτημάτων (4η)



Όταν δεν ισχύει καμία από τις προηγούμενες συνθήκες μας ενδιαφέρουν και τα δύο κλαδιά.



Σπάμε το διάστημα ερωτήματος σε δύο ανεξάρτητα Διαστήματα τα οποία στέλνουμε στα δύο παιδιά.

Interval Trees

```
int query(int QX, int QY, int x, int y, int id) {
    if ( x == QX && y == QY ) return T[id];

        mid = (x + y)/2;
        if ( QY <= mid ) {
            return query(QX, QY, x, mid, 2*id);
        }
        else if ( QX > mid ) {
            return query(QX, QY, mid+1, y, 2*id+1);
        }
        else {
            return max(query(QX, mid, x, mid, 2*id),
                       query(mid+1, QY, mid+1, y, 2*id + 1));
        }
}



    query(QX, QY, 1, N, 1);
```