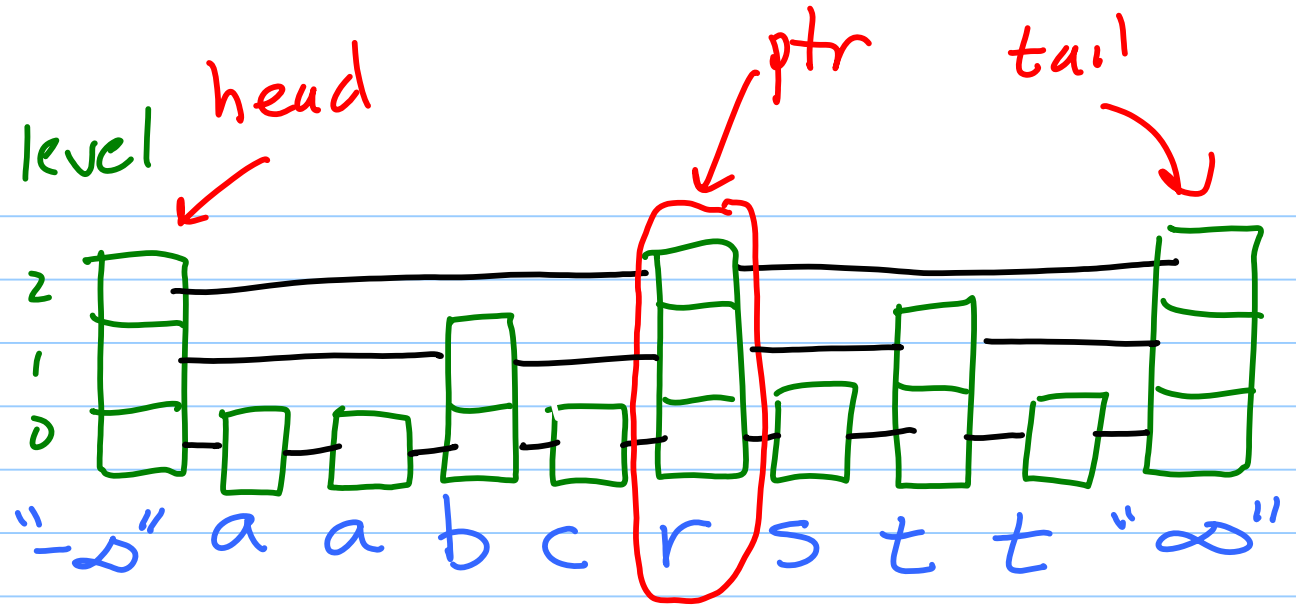


L₂: r
 L₁: b r t
 L₀: a a b c r s t t

Show
 as →



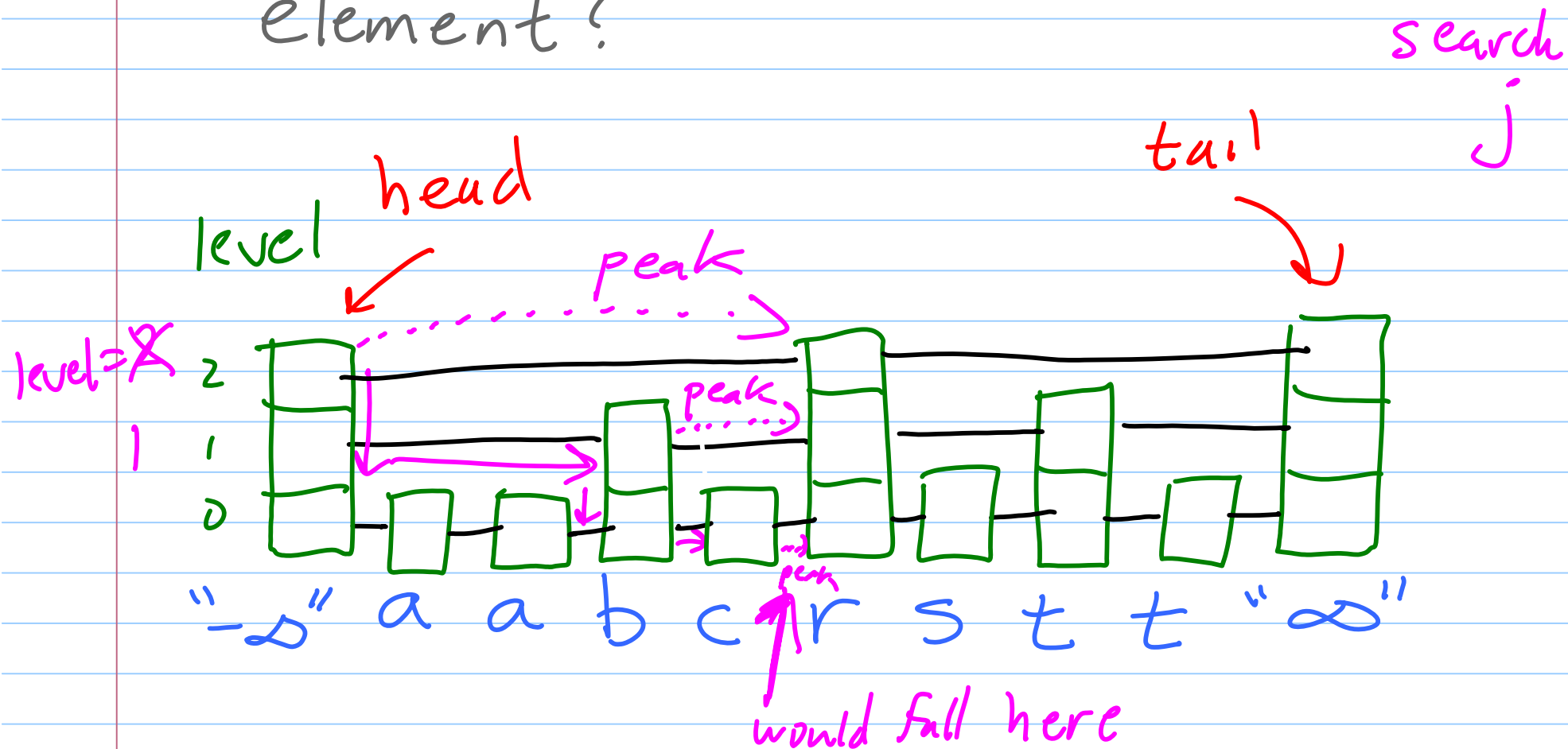
How do you find min? `head.next[0]`

max? `tail.prev[0]`

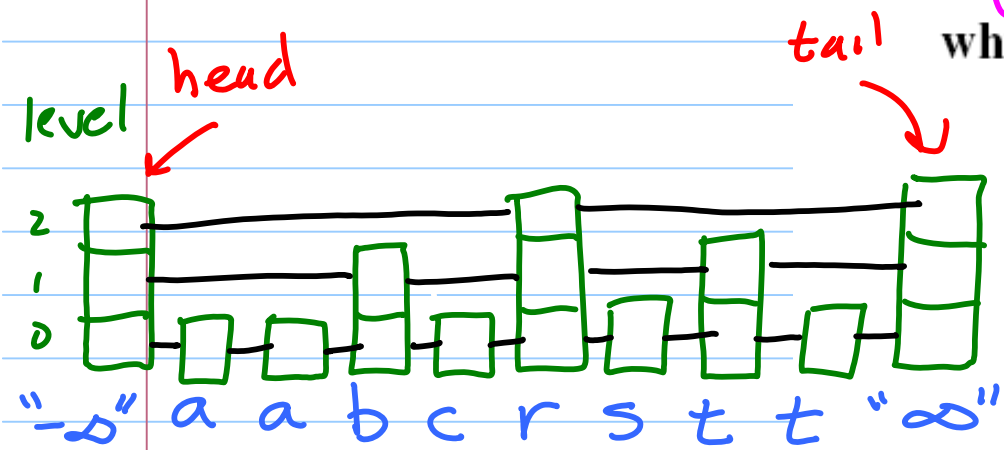
How do you move to next/prev element from a given tower?

`ptr.next[0]` `ptr.prev[0]`

How do you search for an element?



L₂: r
 L₁: b r t
 L₀: a a b c r s t t



```
Tower<E> findFirstOccurrence(E target) {
```

```
    Tower<E> left = head;
```

look in for

```
    Tower<E> right = tail;
```

```
    int level = height - 1;
```

```
    while (level ≥ 0) {
```

```
        Tower<E> next = left.next(level);
```

peak ahead

```
        if (right == next)
```

```
            level--;
```

drop down

```
        else {
```

```
            if (comp.compare(target, next.element) > 0)
```

```
                left = next;
```

target > next

```
            else {
```

move forward

```
                right = next;
```

```
                level--;
```

drop down + reset right

```
            }
```

```
        }
```

```
    }  
    return right;
```

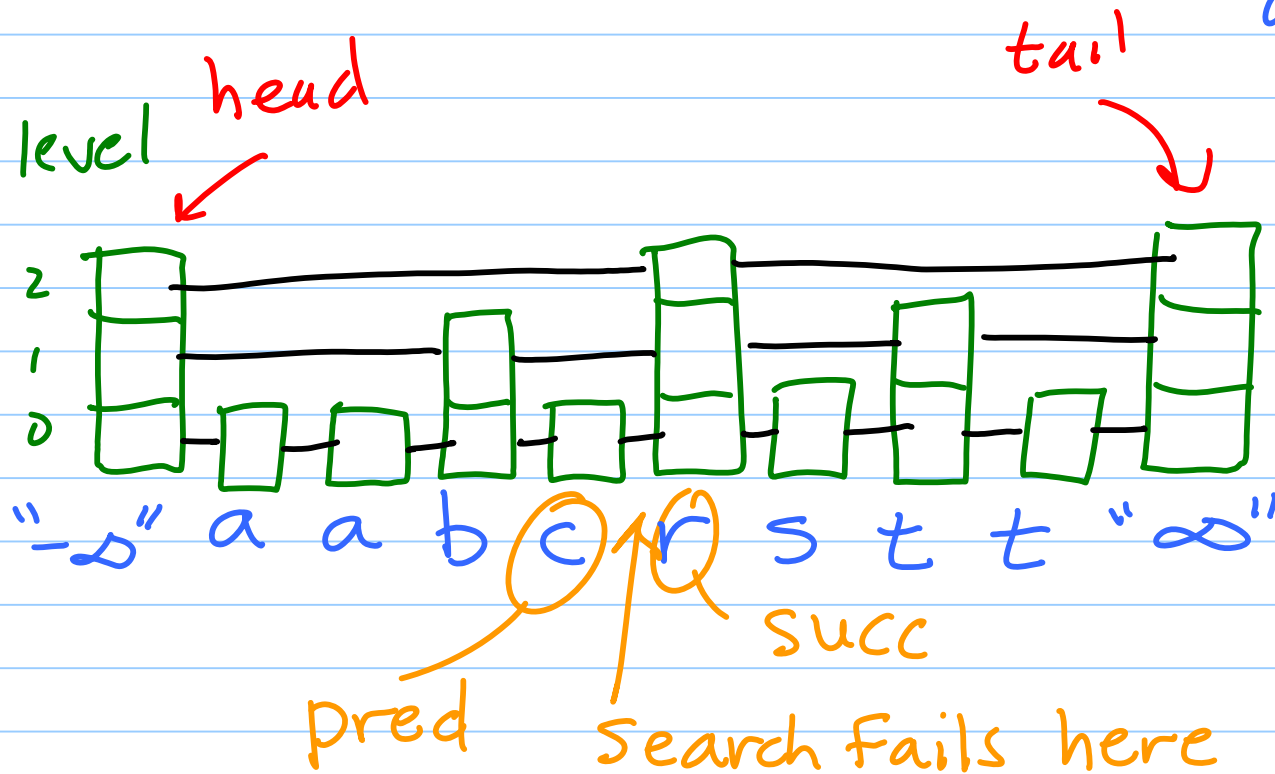
first occurrence or where to insert

```
}
```

Search for t

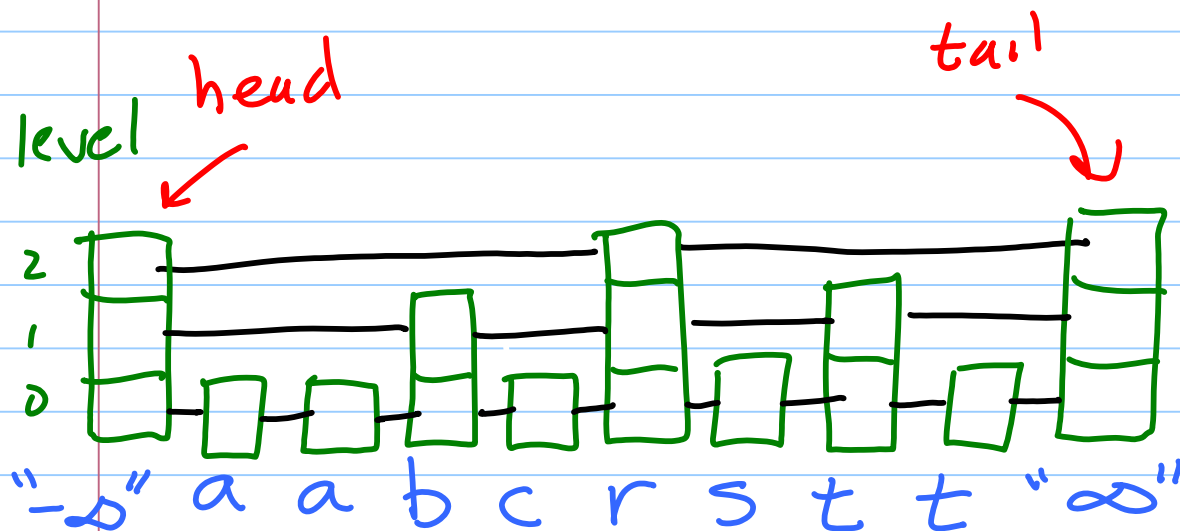
Search for d

How can you find predecessor/successor of an element (e.g. d)?



Use level 0 to navigate in sorted order

Insertion



Let p be
prob of a
tail (continue).

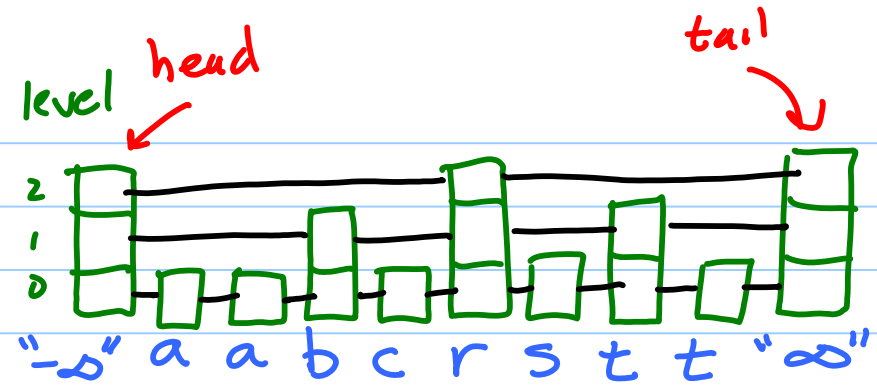
Ex. $p = 1/4$

Find position to insert using search
Then randomly pick height of new tower
as # biased coin flips until head obtained

Once the position is found and height is selected, just splice new tower into list for each level (already know where from search).

Also update height (+ occasionally must resize head/tail).

Deletion



Search to find \pm then splice out of each level it is in (just like in a linked list).

Also update height (\pm occasionally must resize head/tail).