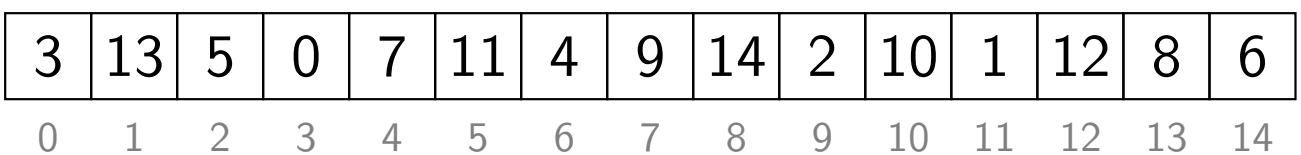
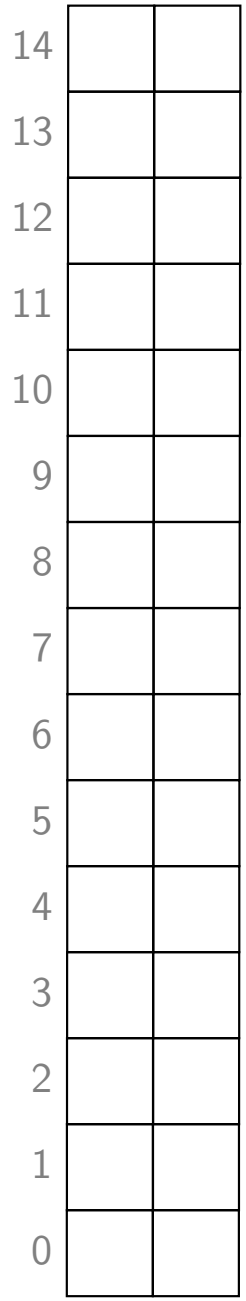
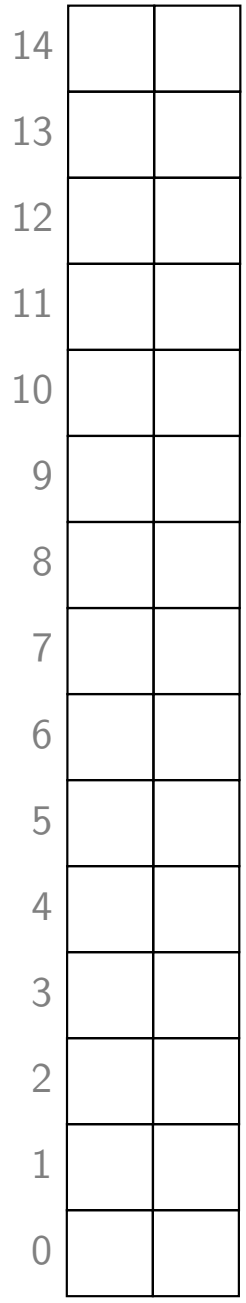
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



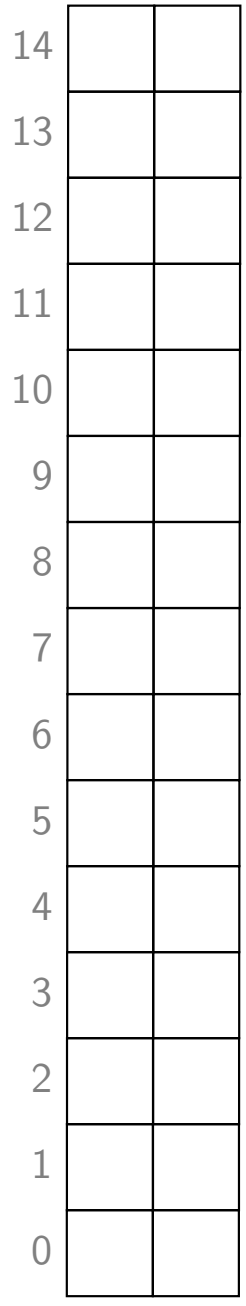
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

| 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



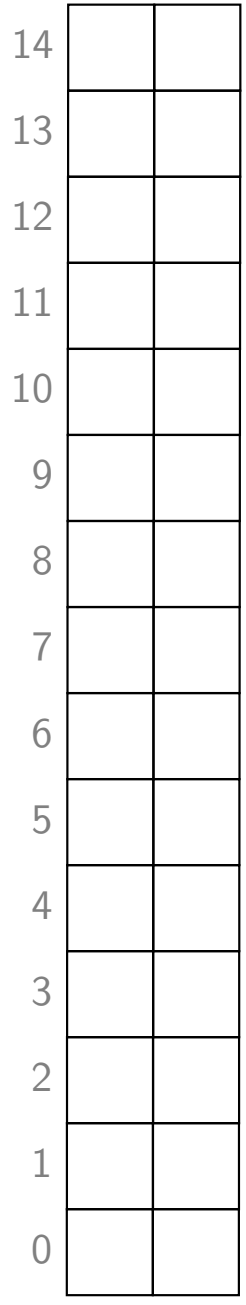
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

| 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

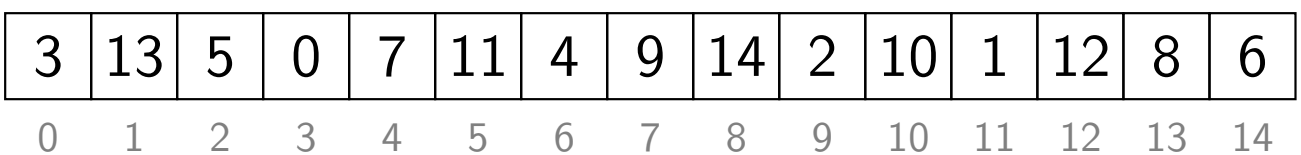


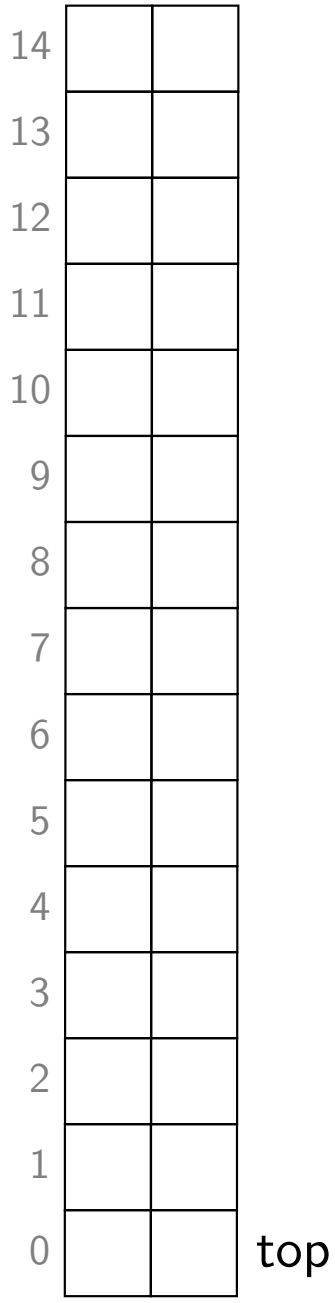
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

| 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

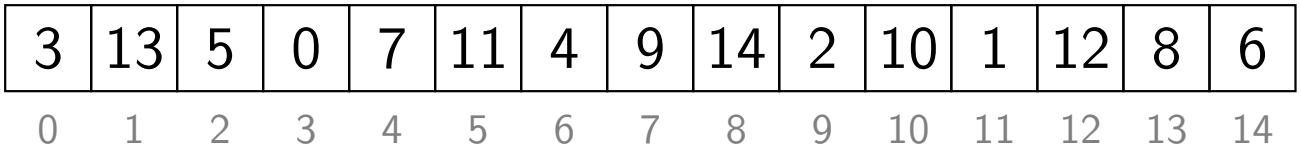


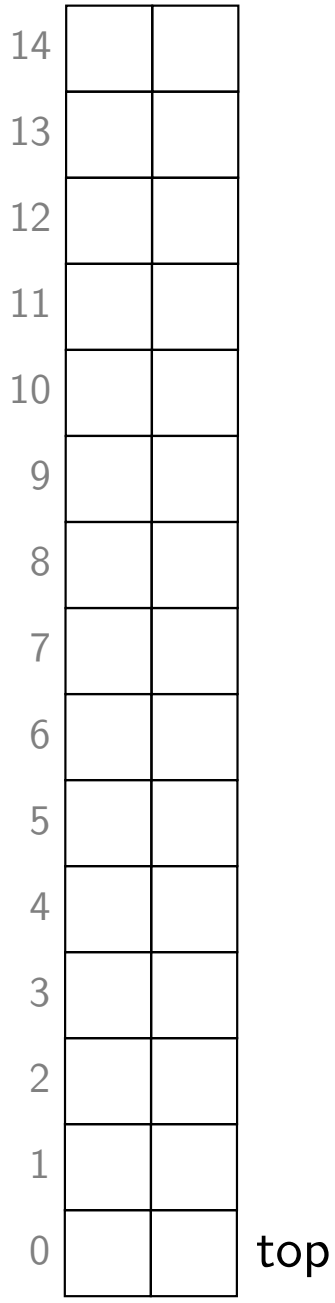
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



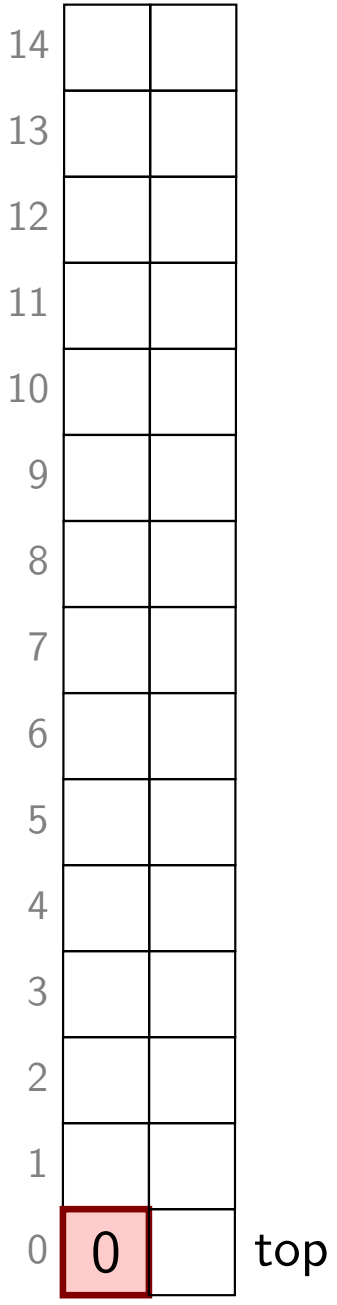
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



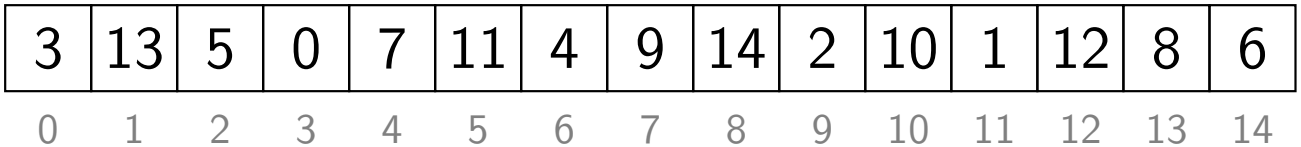



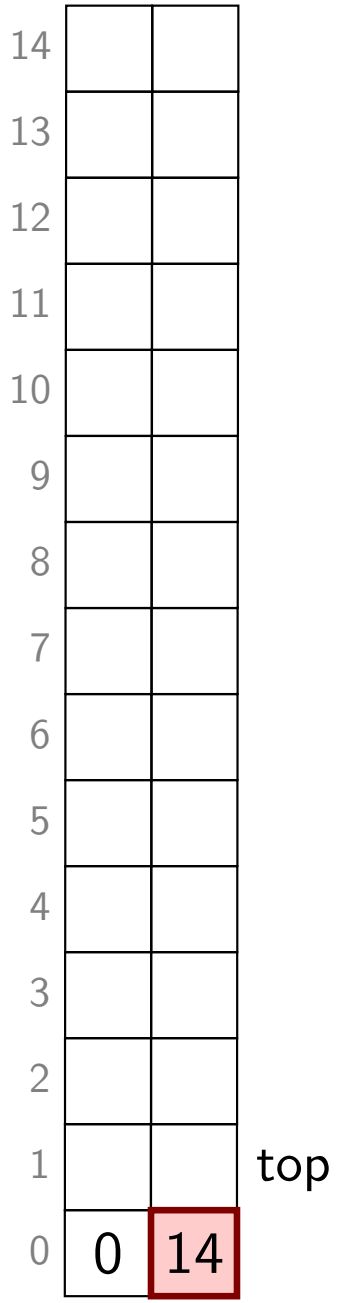
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

| 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



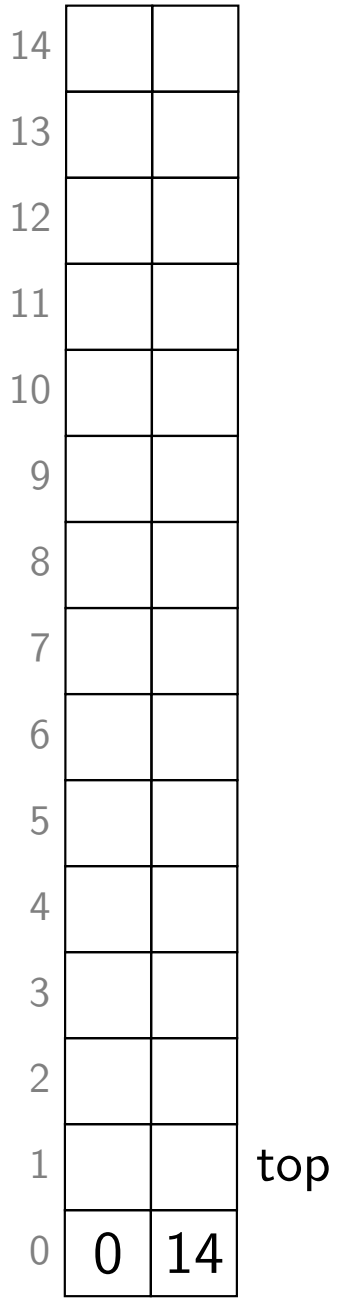
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

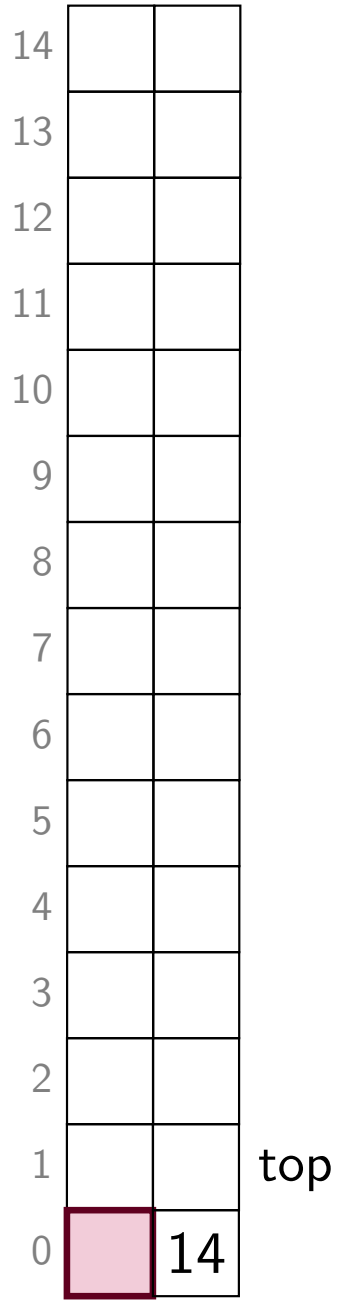
| 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



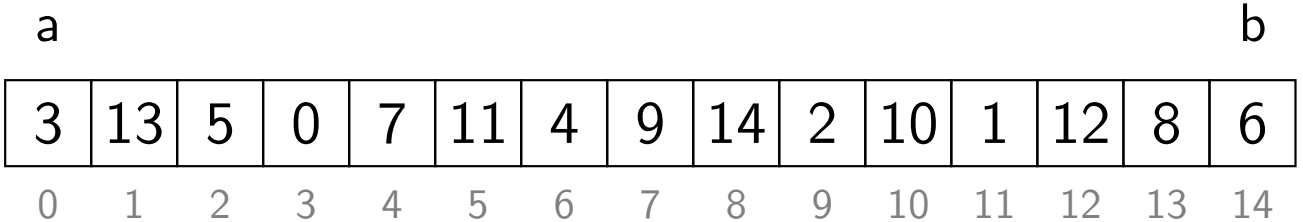
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

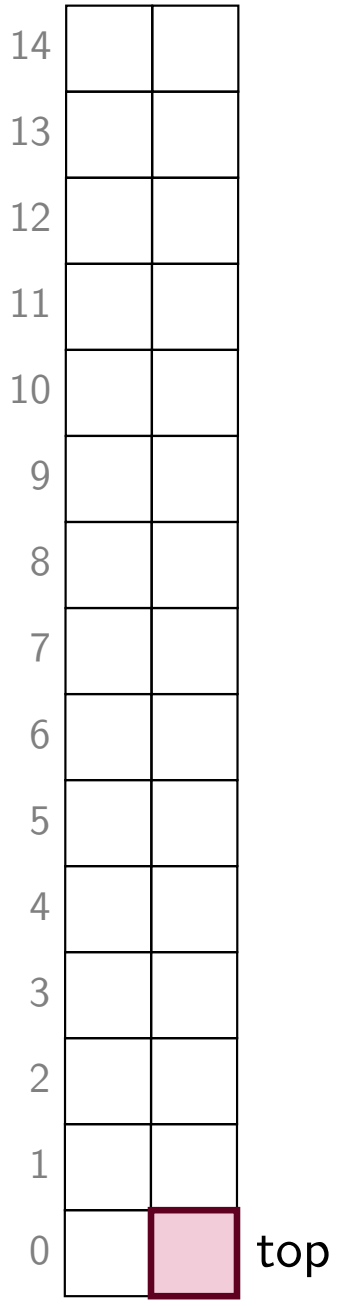
a

| 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

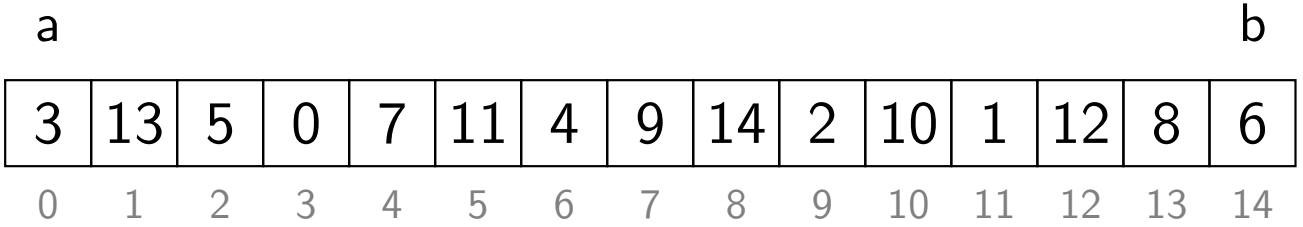


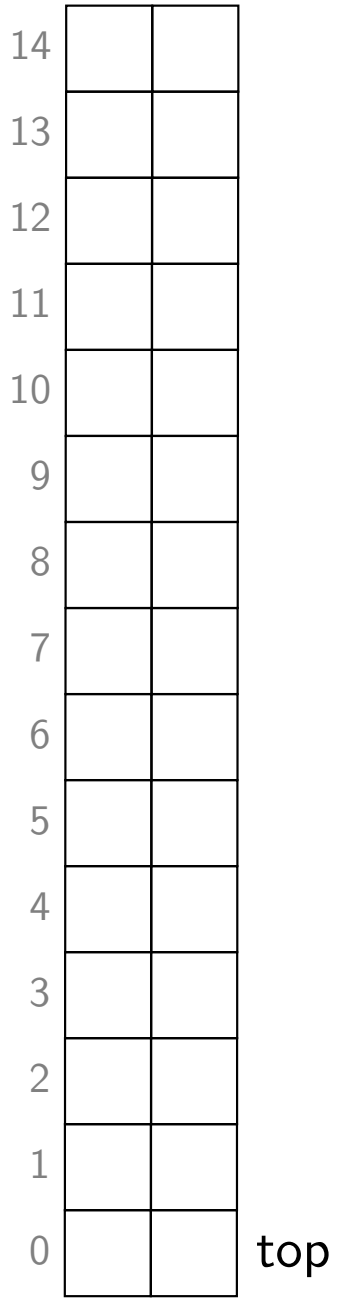
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



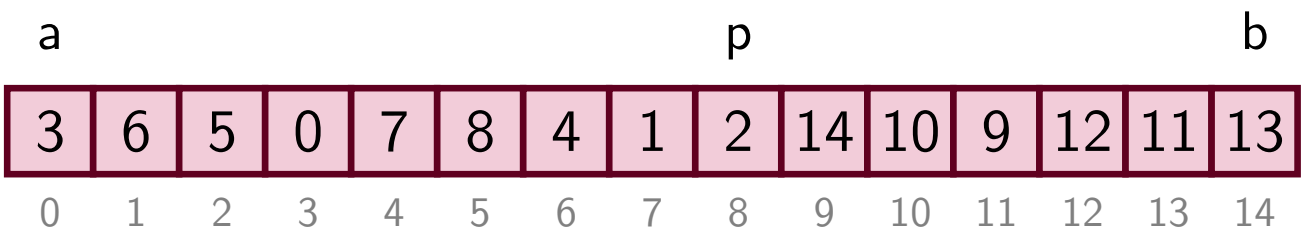



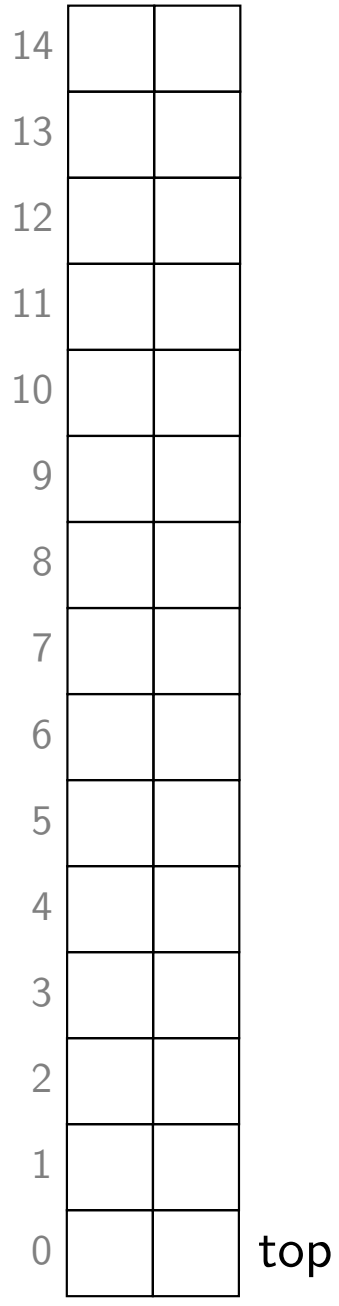
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



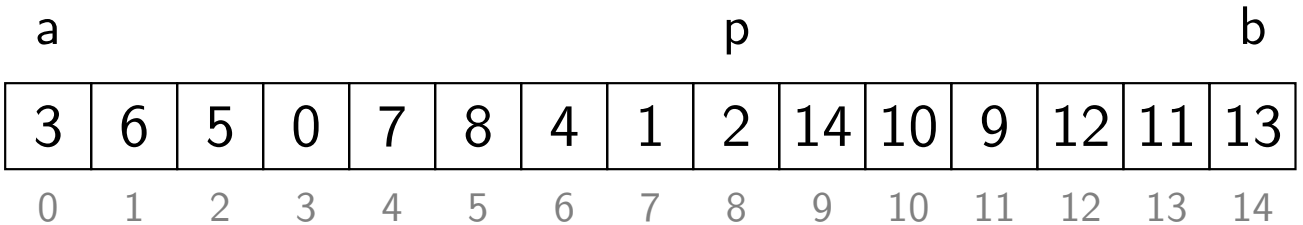



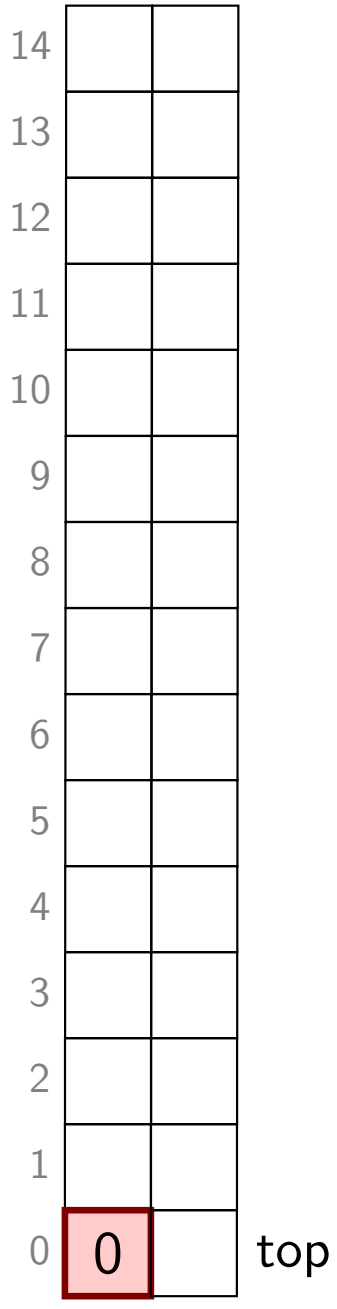
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



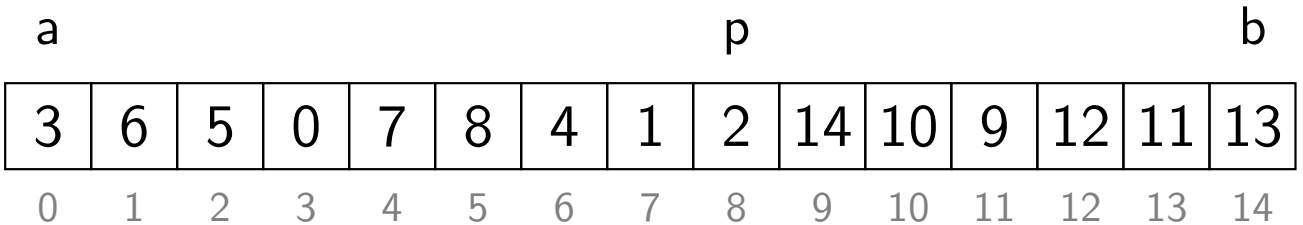



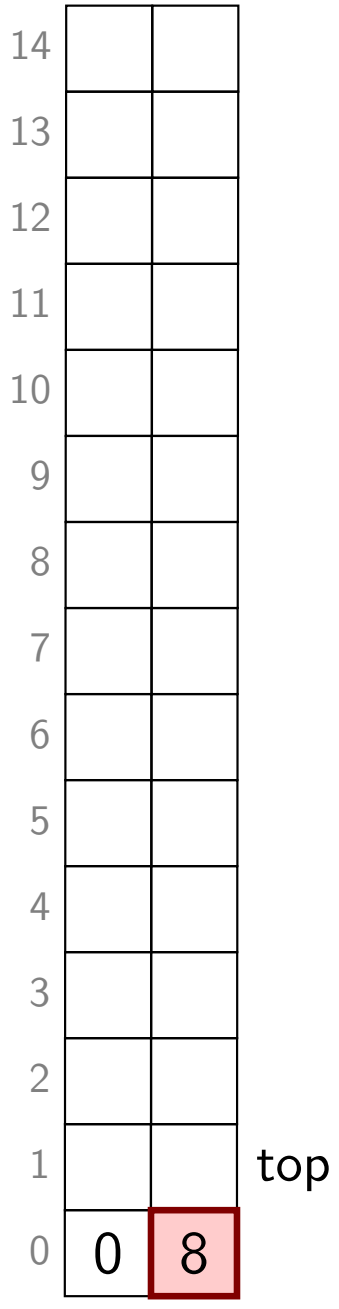
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



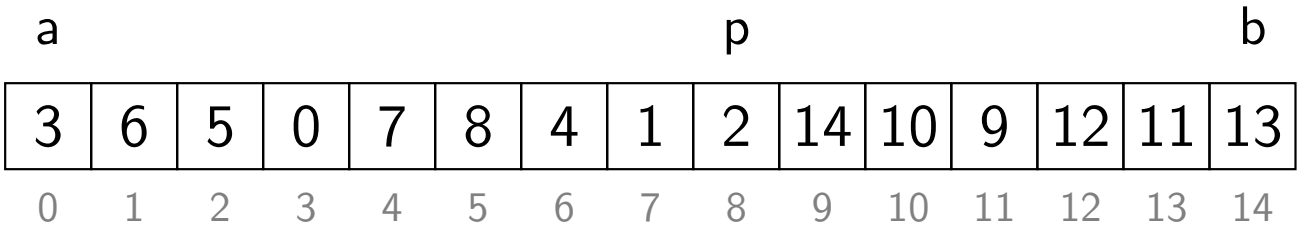



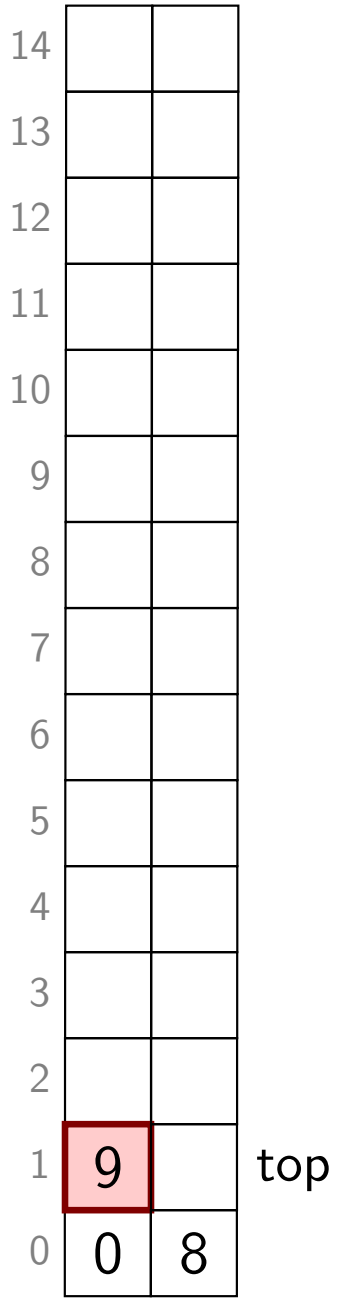
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



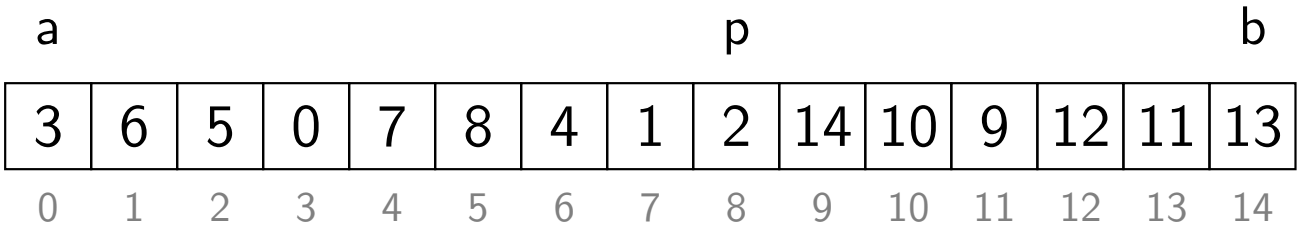



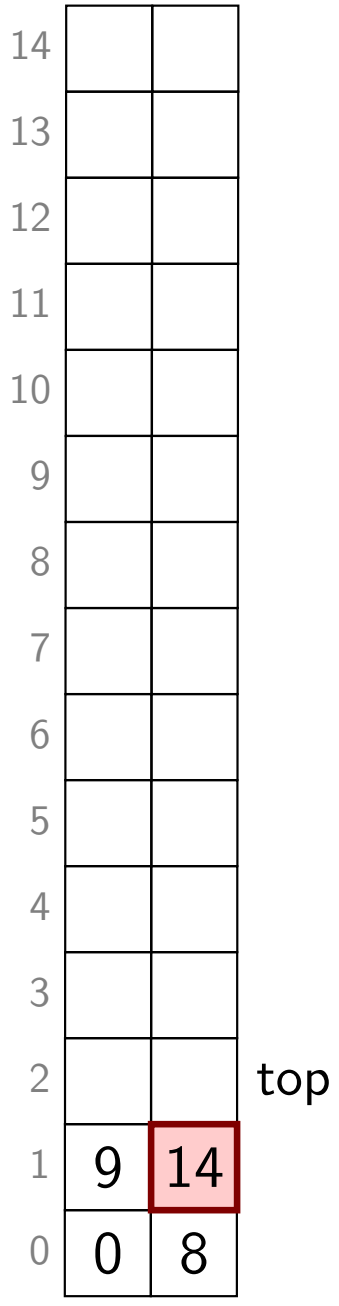
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



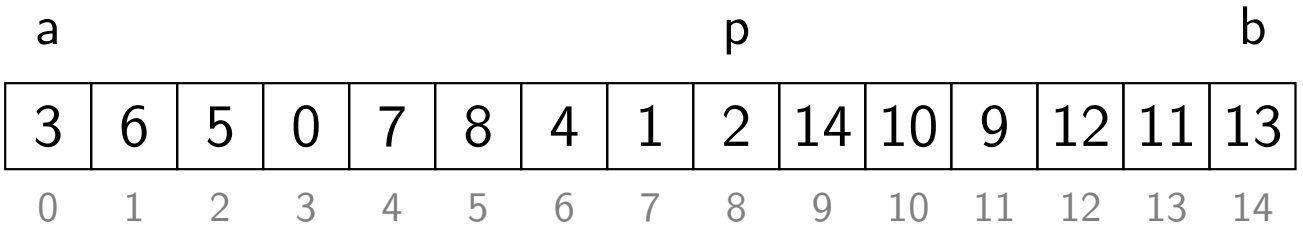



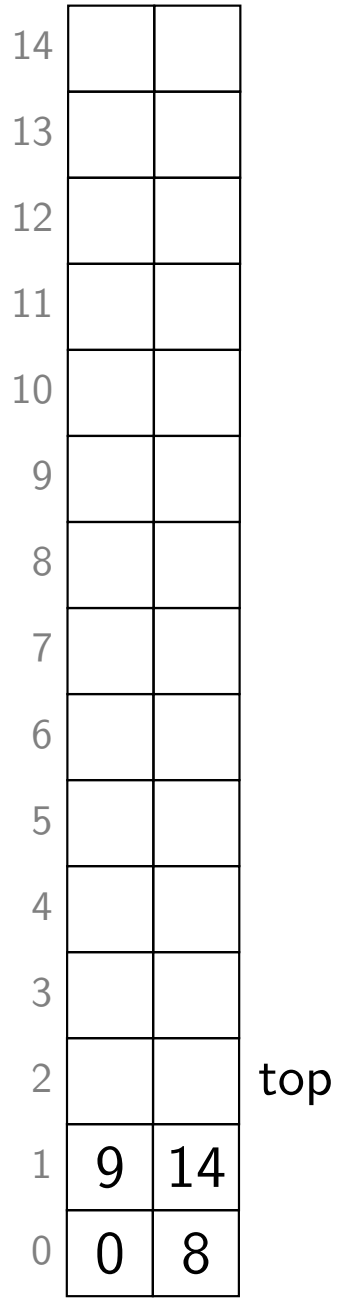
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



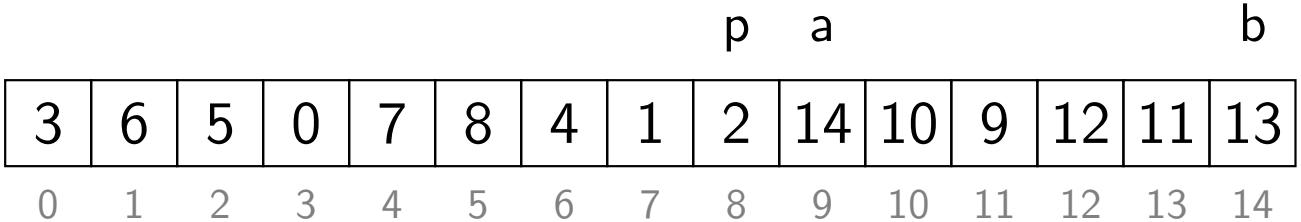



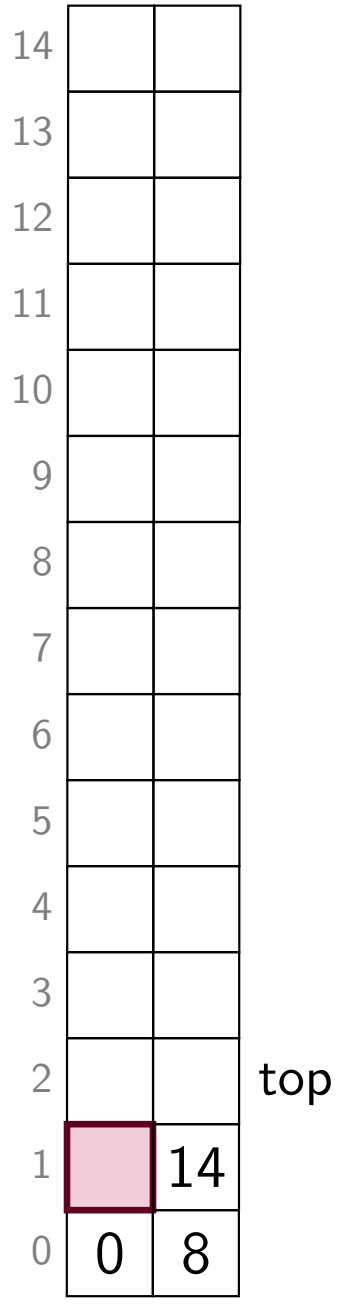
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



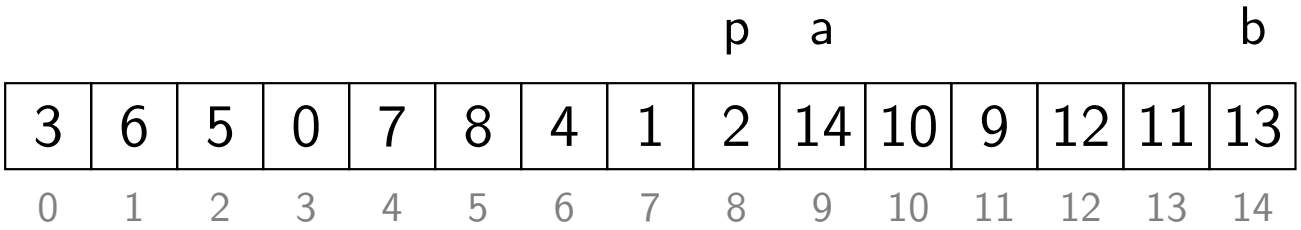



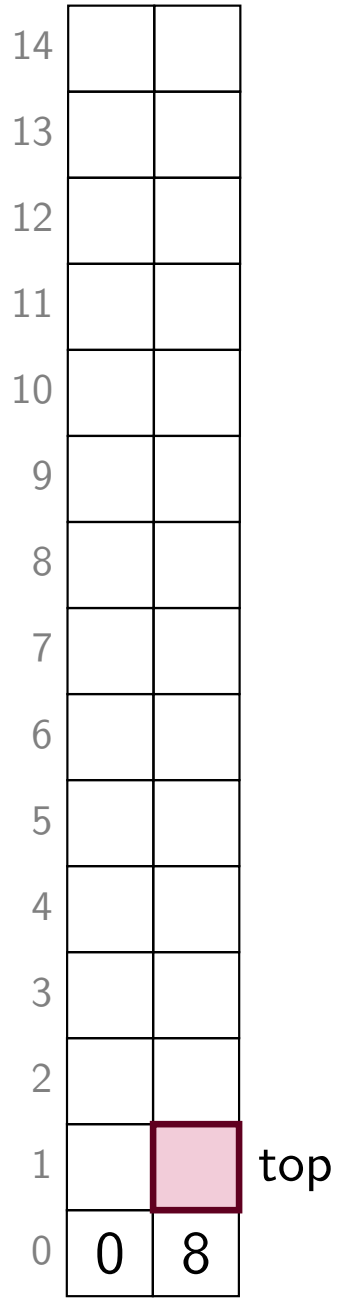
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



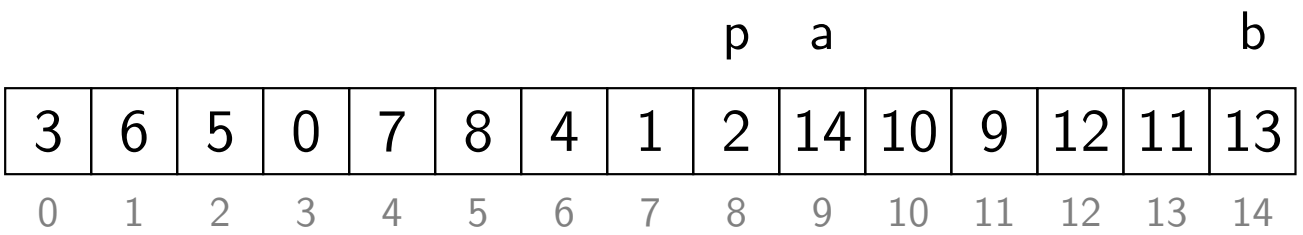



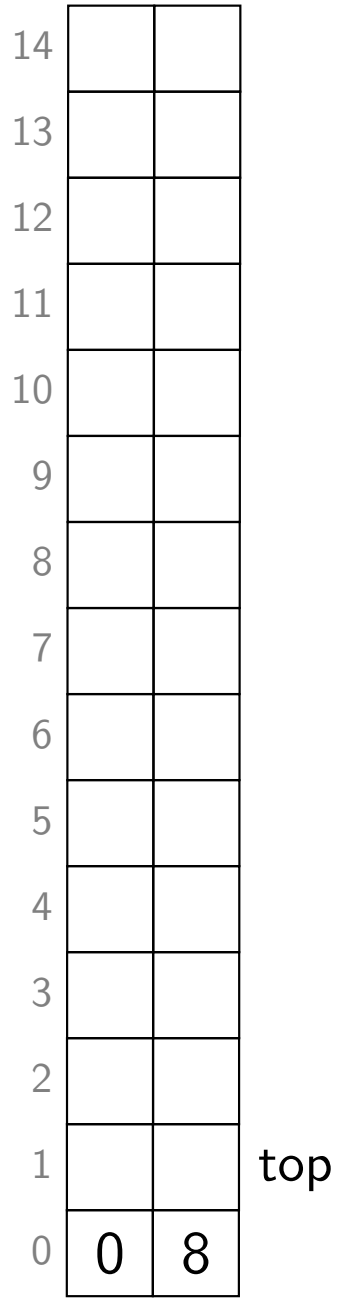
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



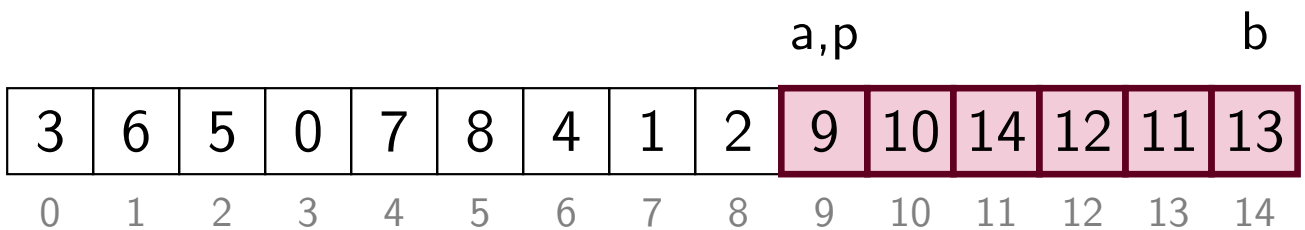



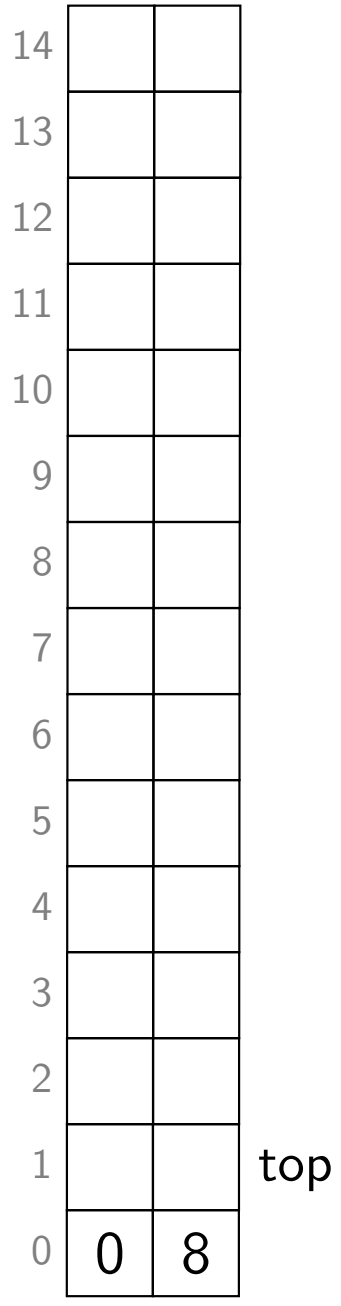
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



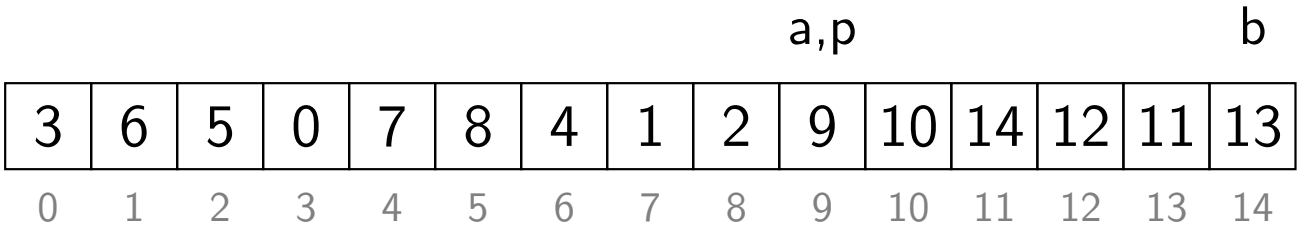



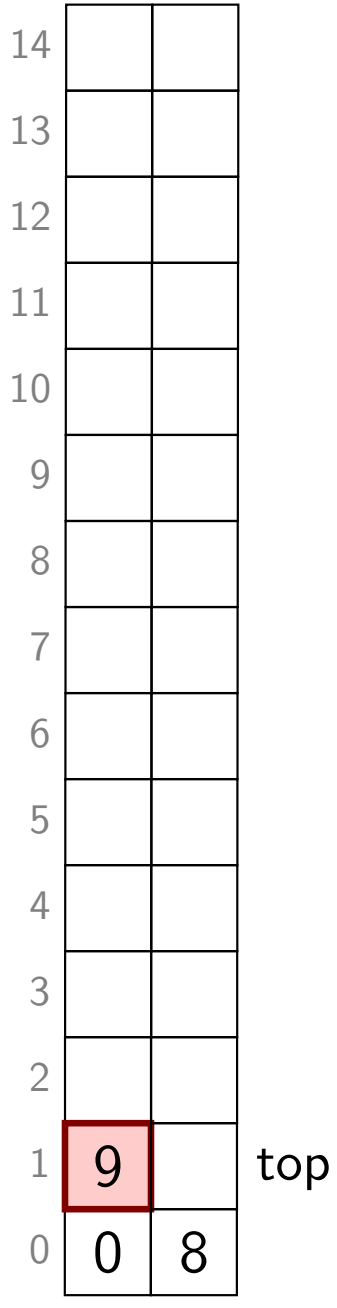
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



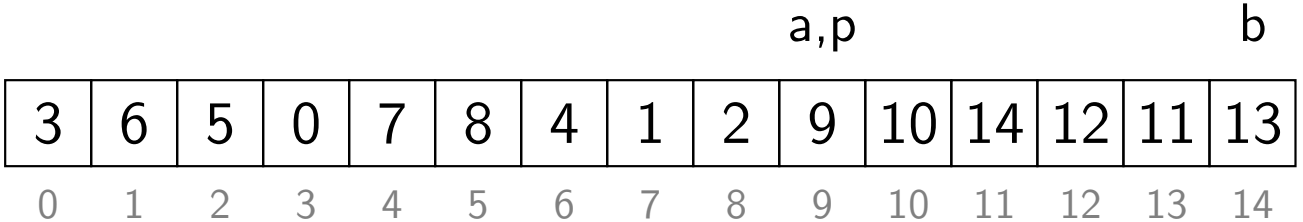



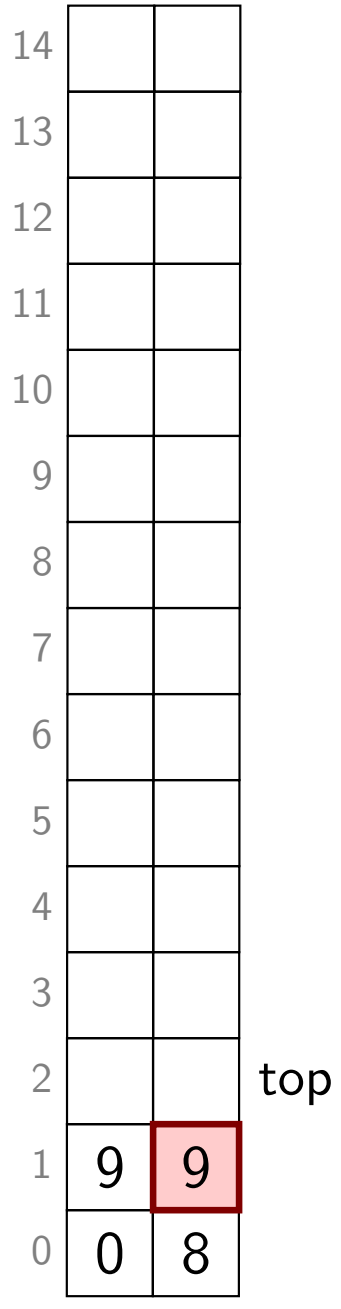
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



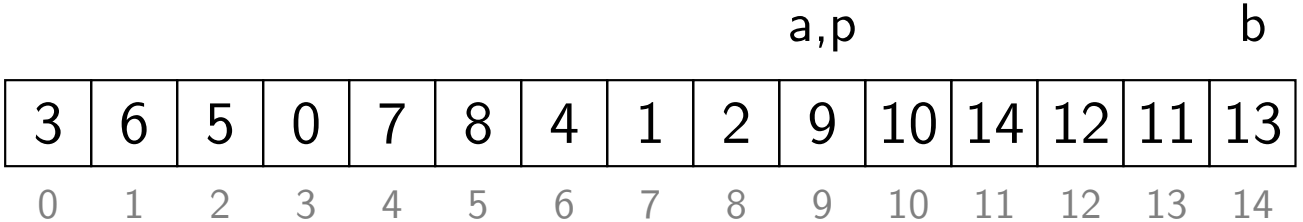



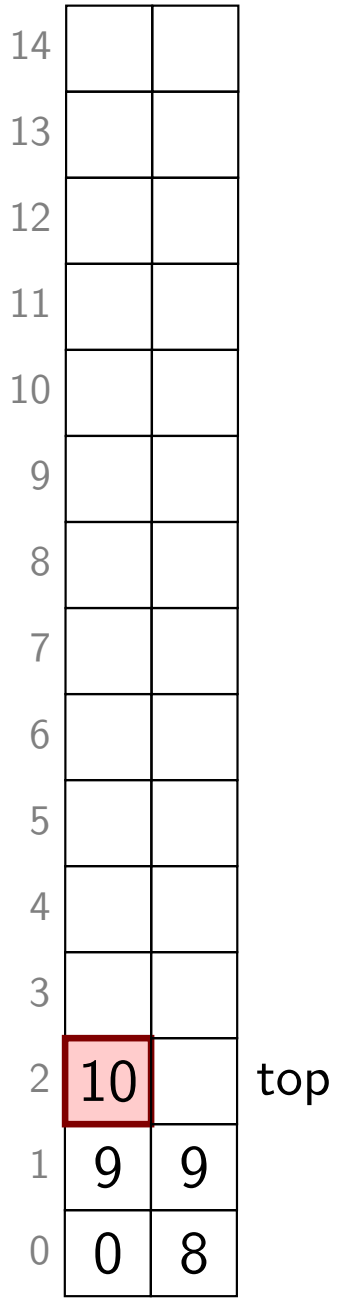
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



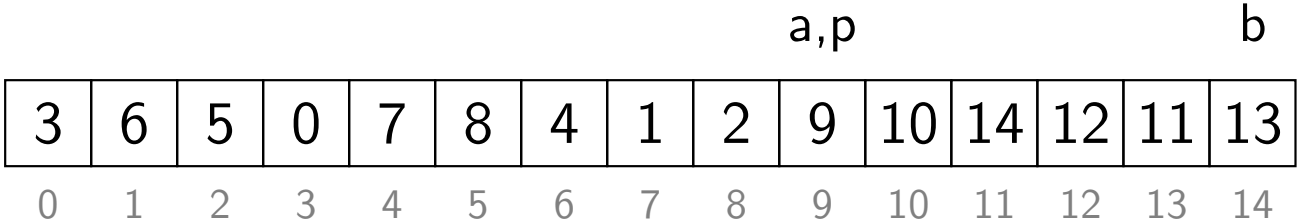



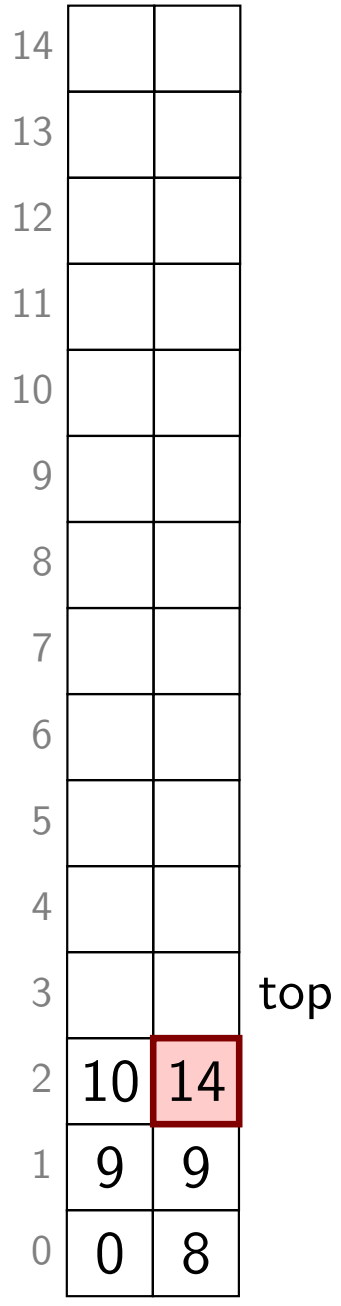
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



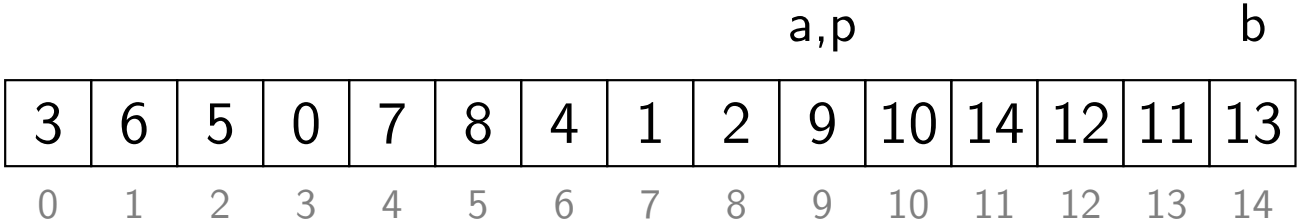



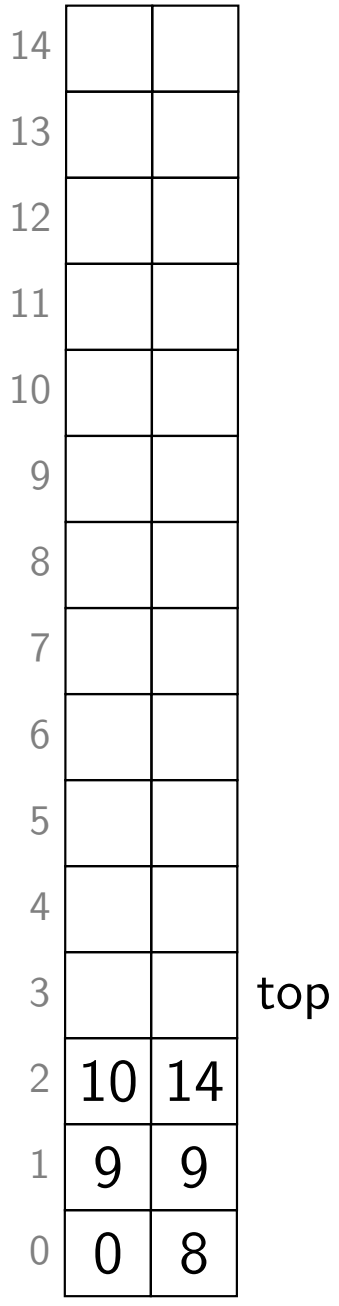
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



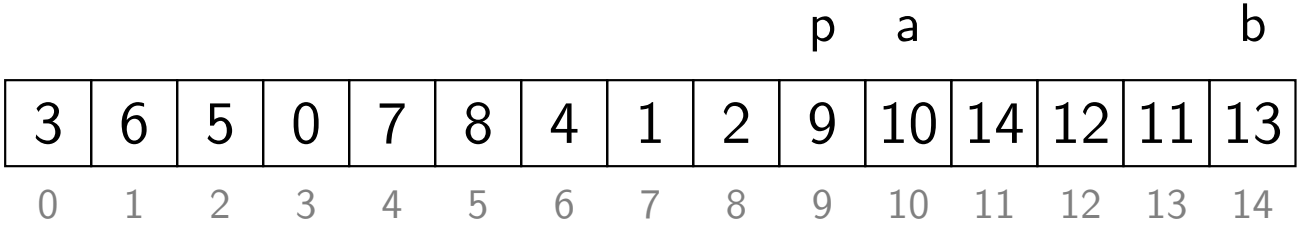



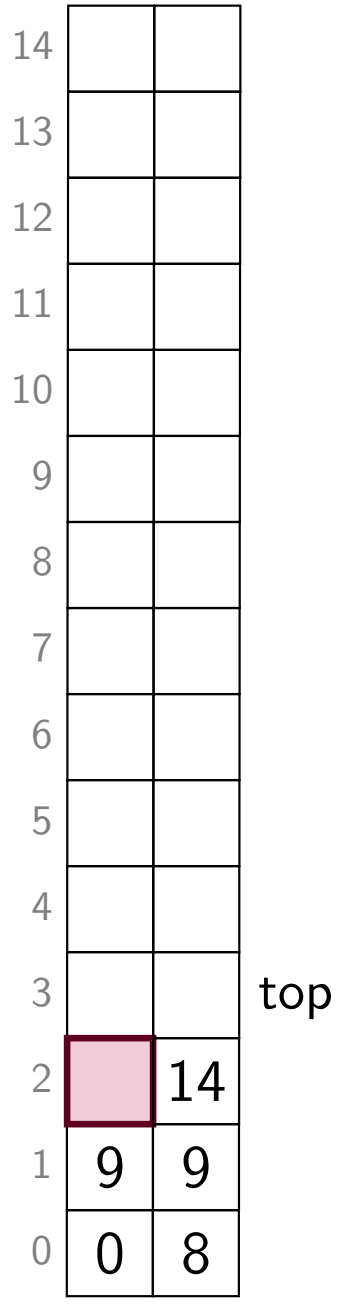
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



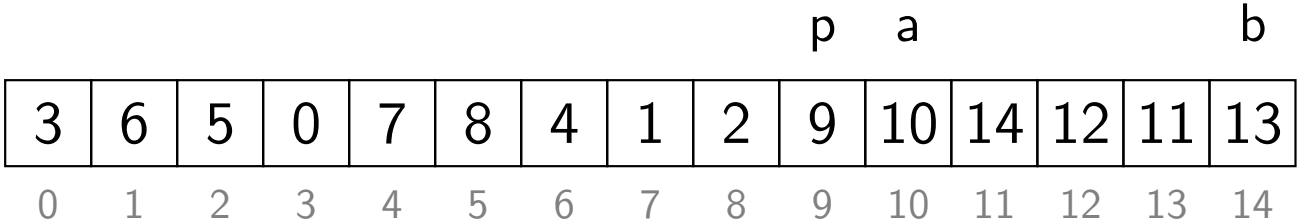



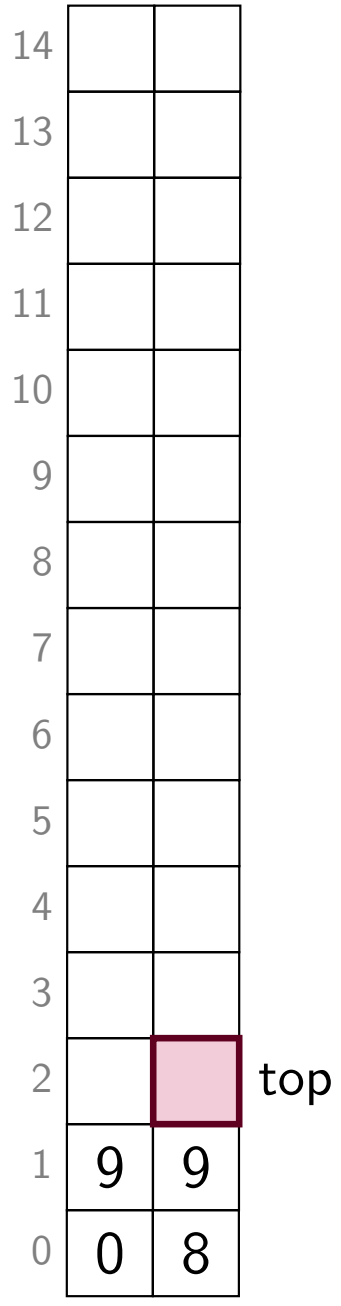
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



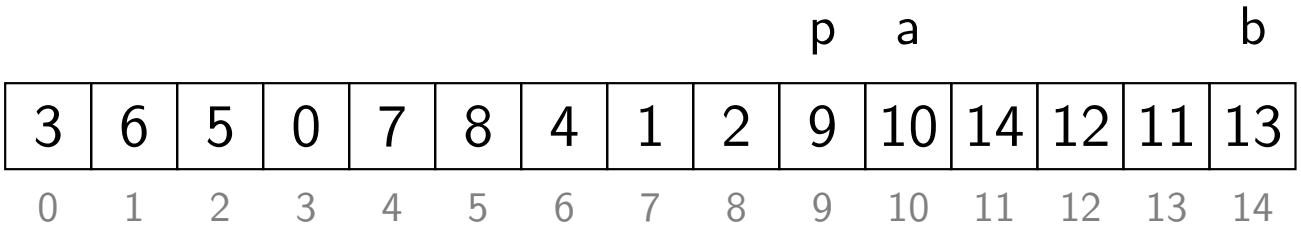



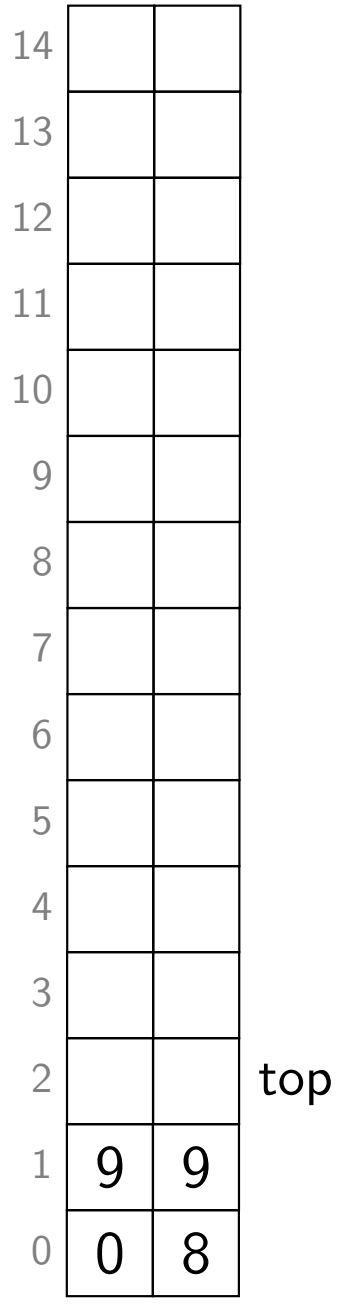
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



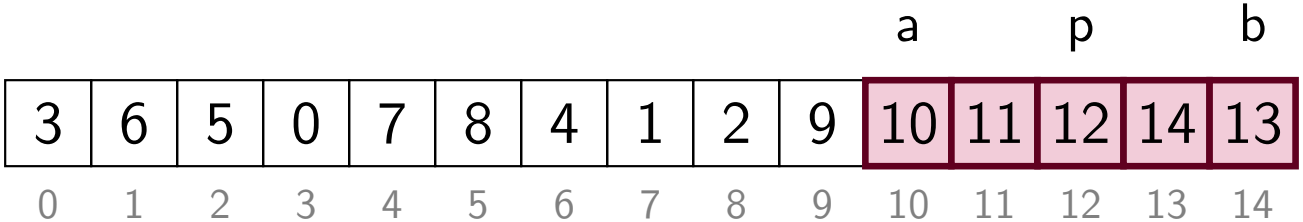



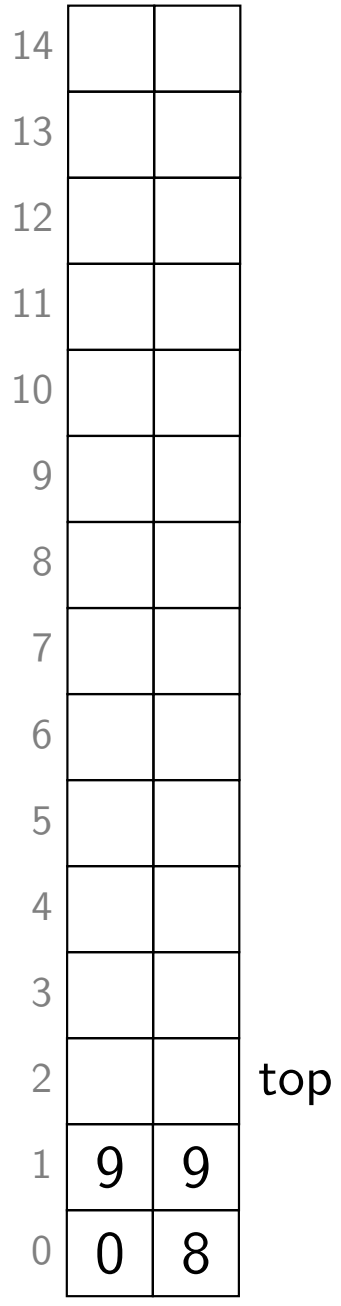
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



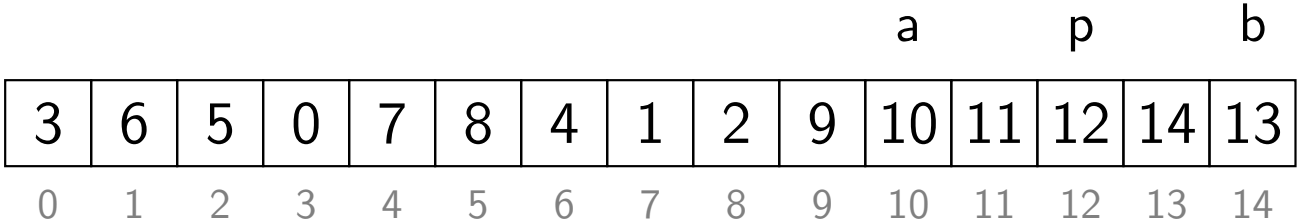



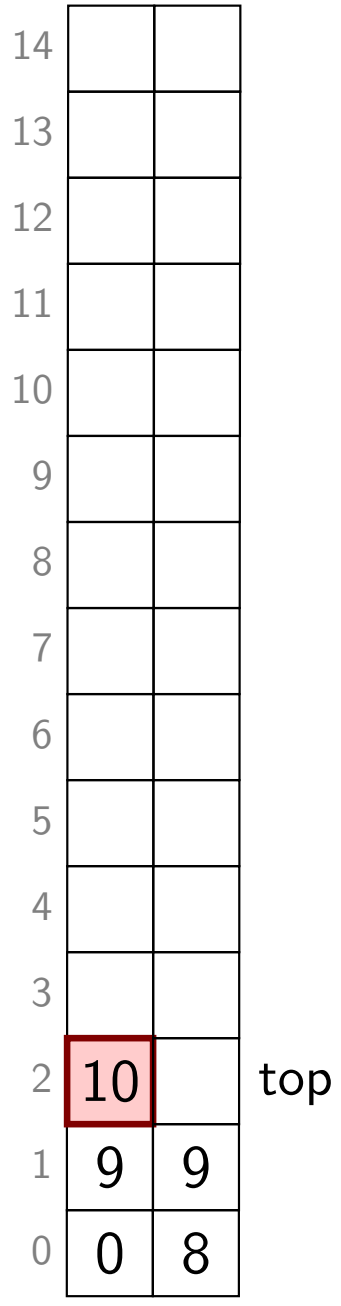
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



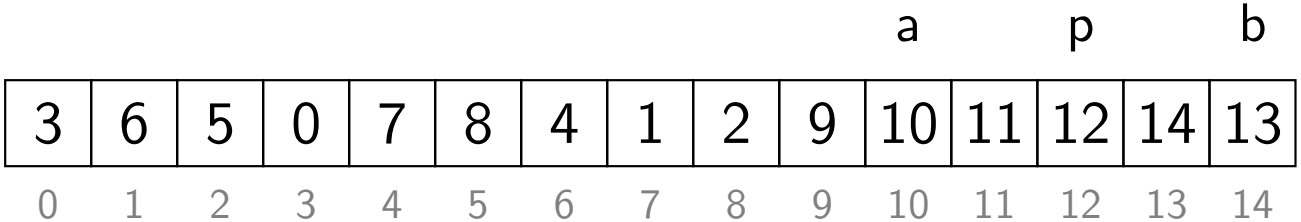



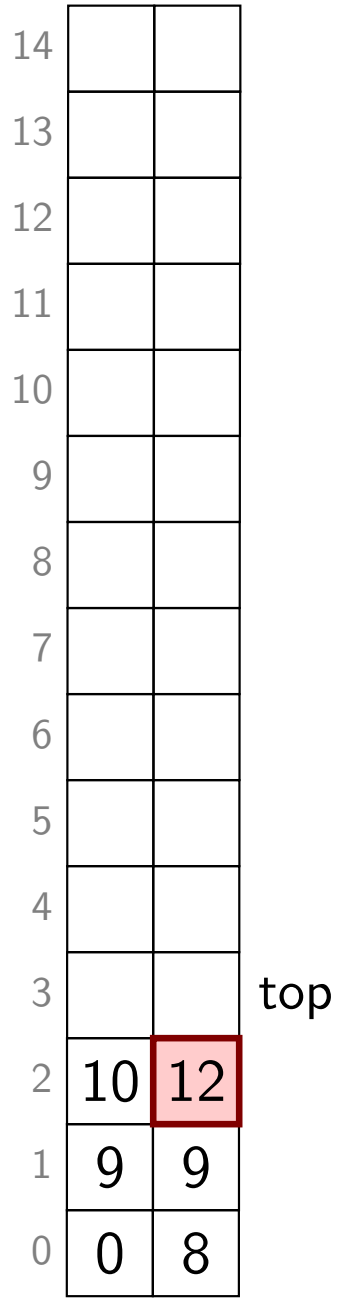
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



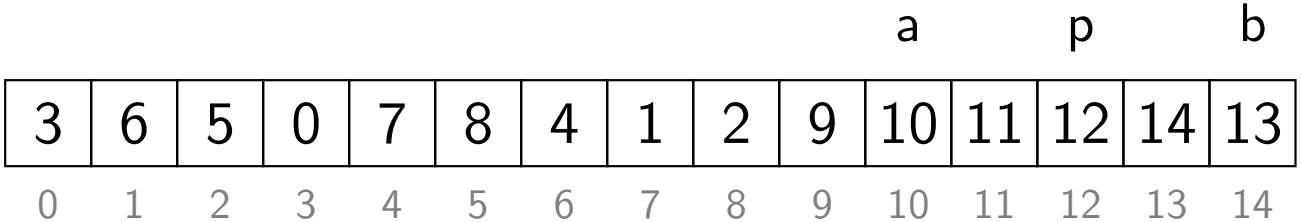



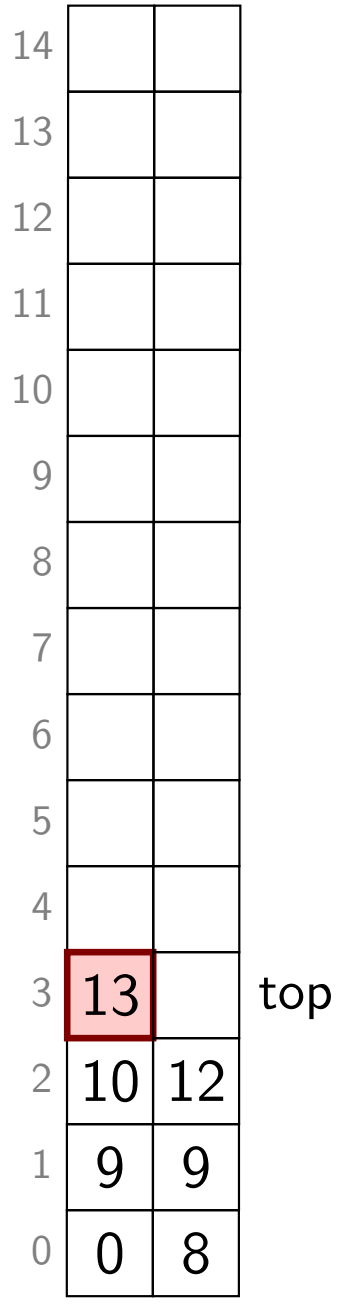
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



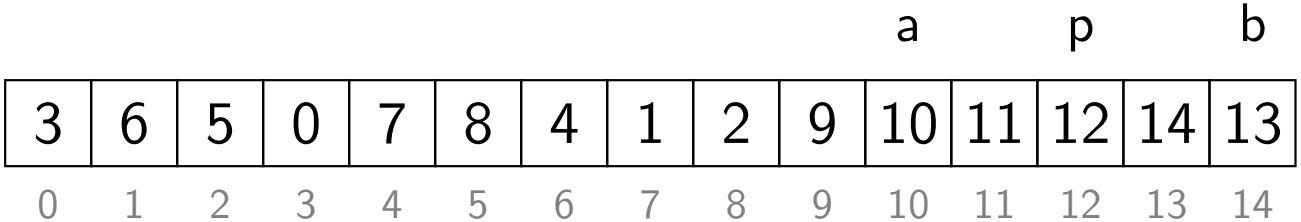



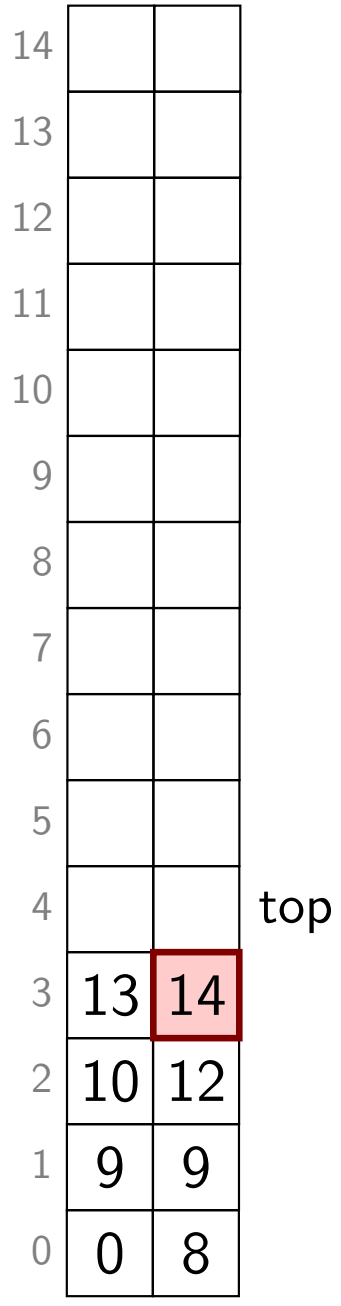
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



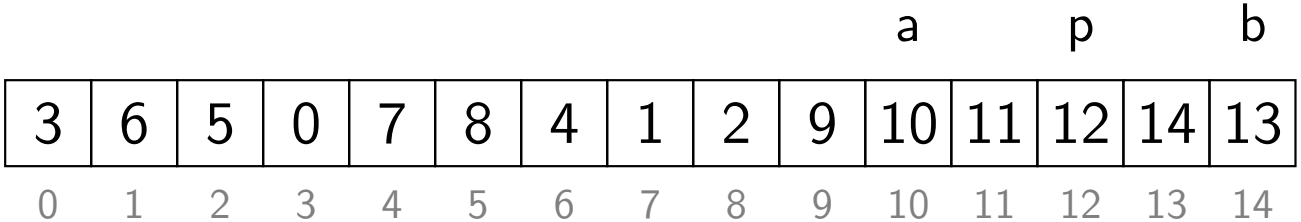



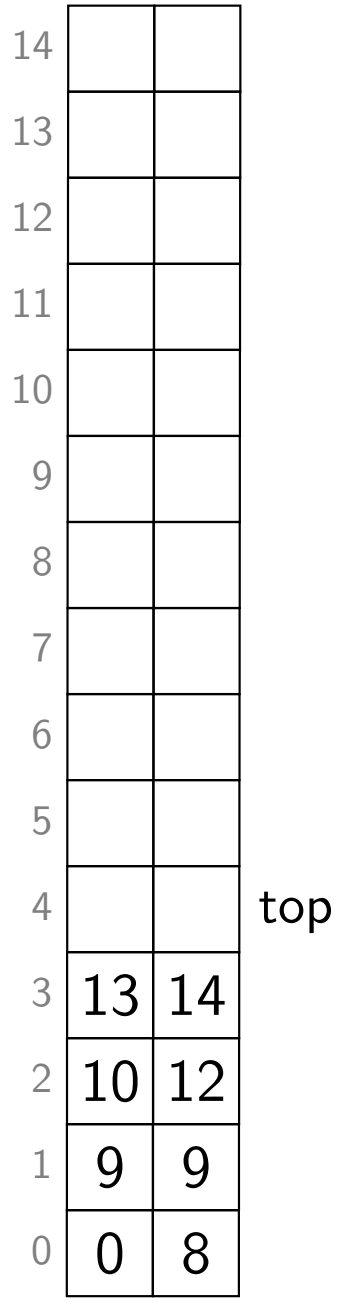
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



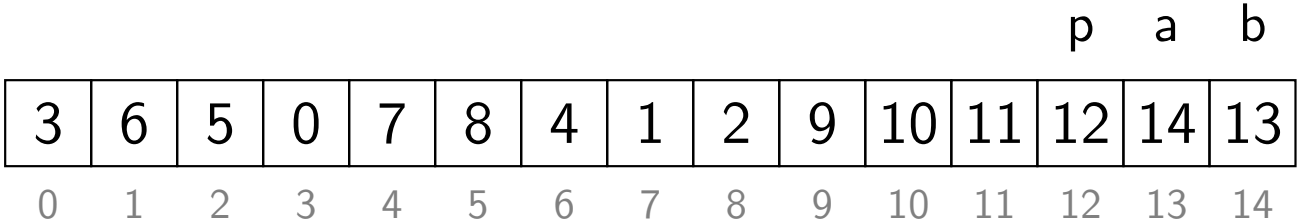



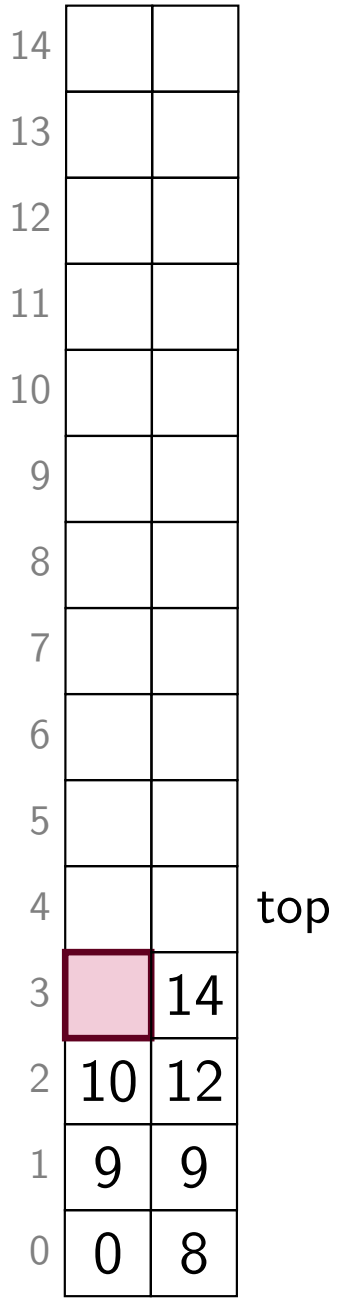
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



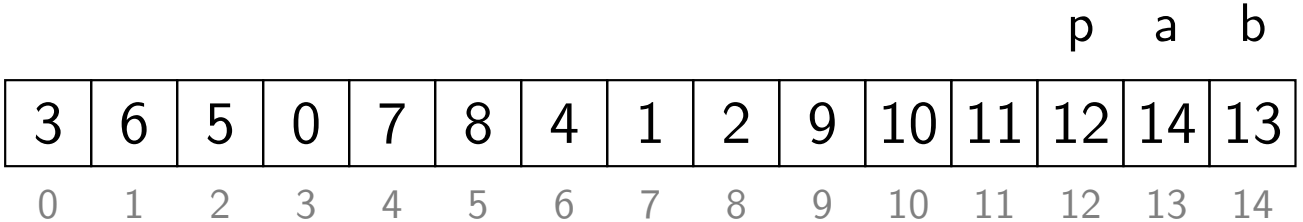



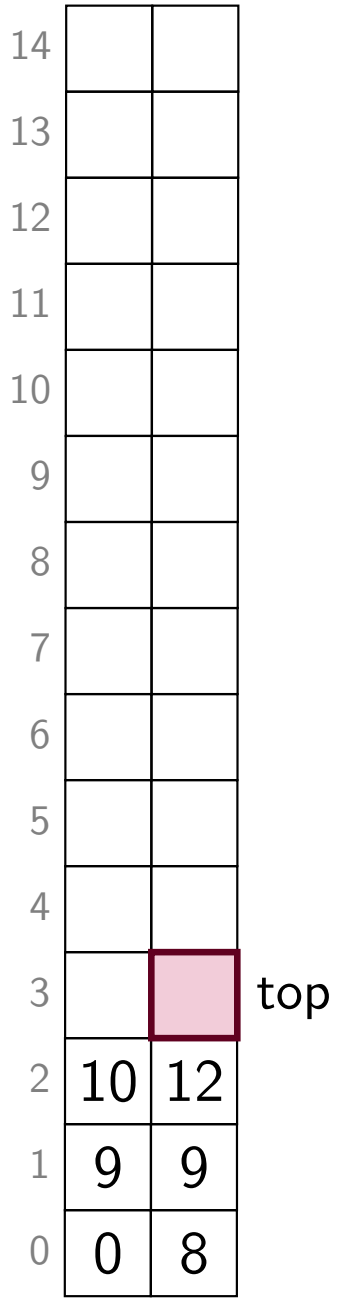
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



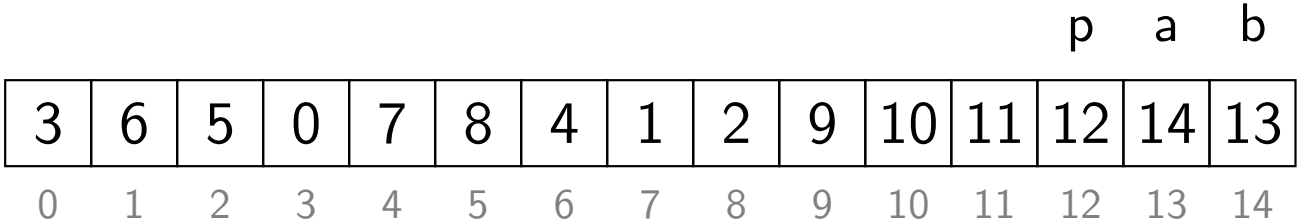



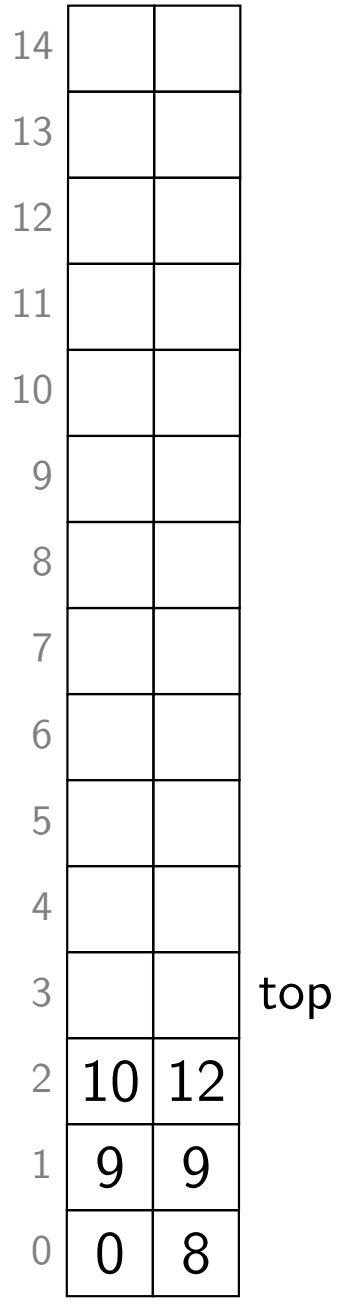
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



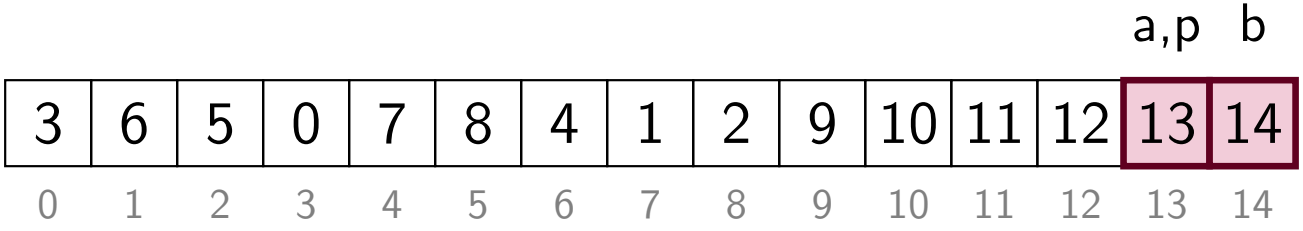



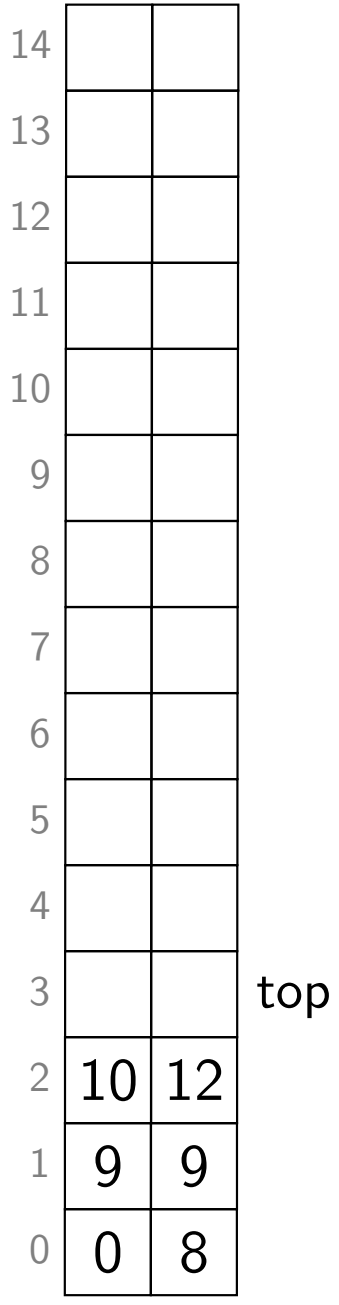
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



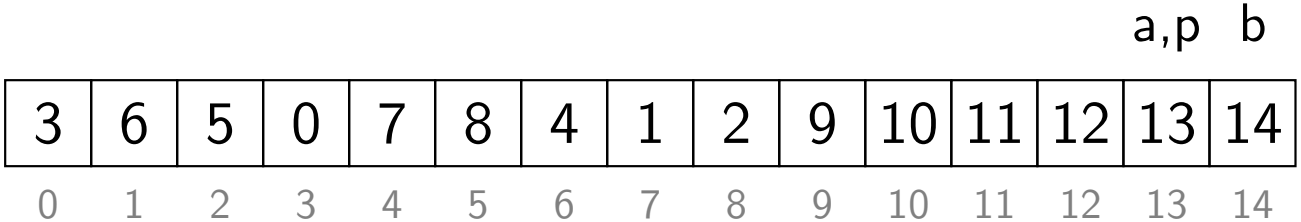
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

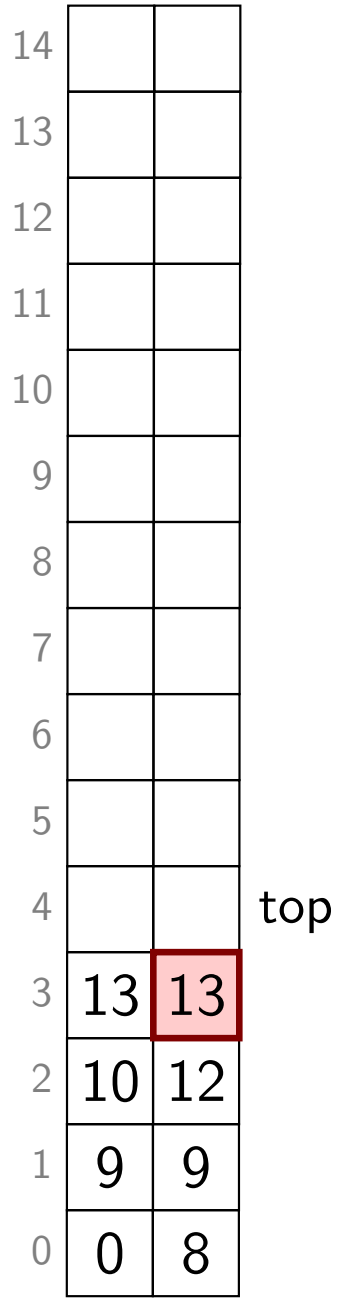
a,p b

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

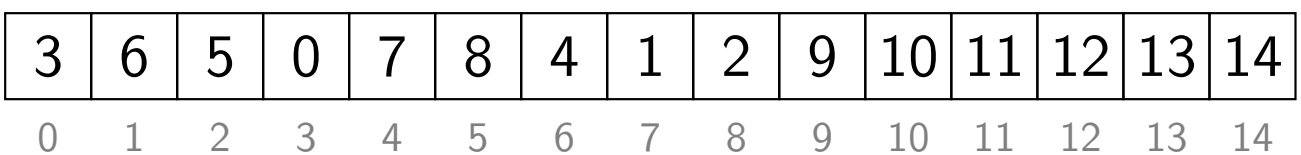
| | | | |
|---|---|---|---|
| 14 | | | |
| 13 | | | |
| 12 | | | |
| 11 | | | |
| 10 | | | |
| 9 | | | |
| 8 | | | |
| 7 | | | |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | 13 | | top |
| 2 | 10 | 12 | |
| 1 | 9 | 9 | |
| 0 | 0 | 8 | |

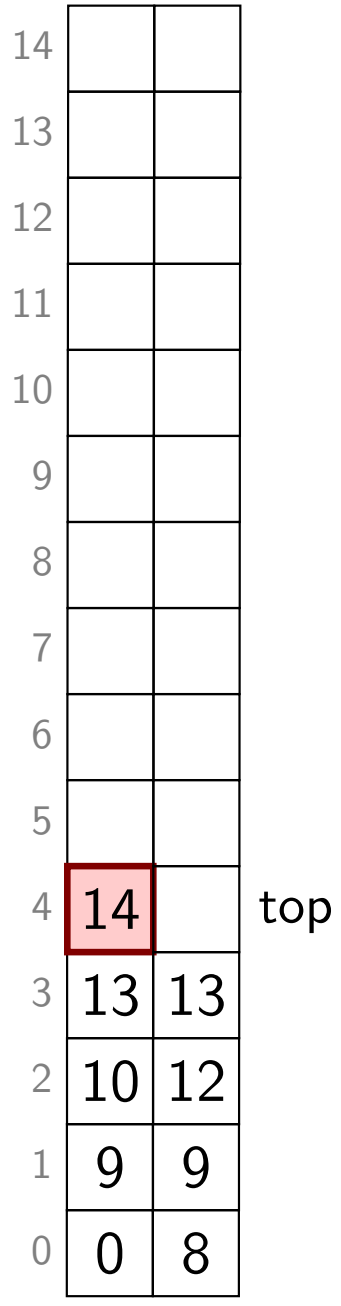
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

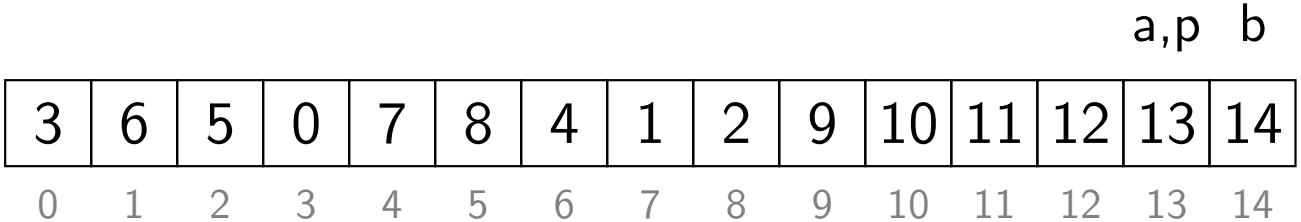
a,p b

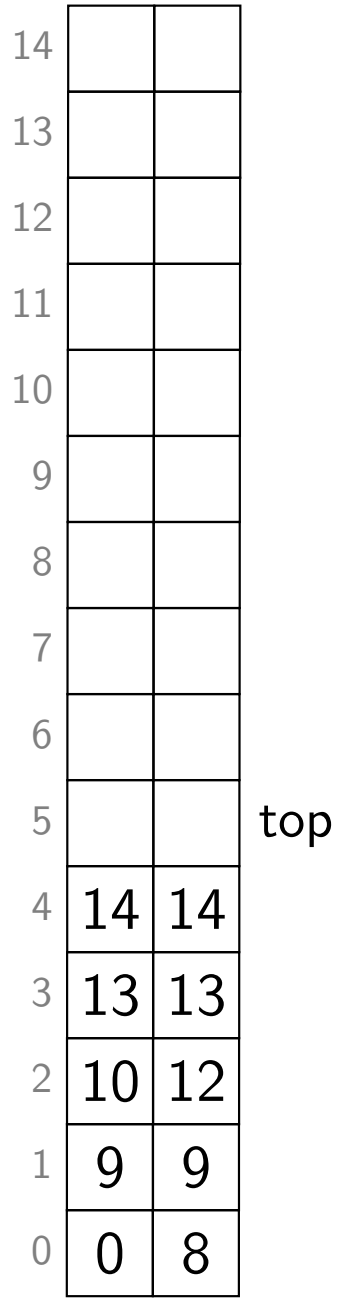
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| | | |
|---|---|---|
| 14 | | |
| 13 | | |
| 12 | | |
| 11 | | |
| 10 | | |
| 9 | | |
| 8 | | |
| 7 | | |
| 6 | | |
| 5 | | |
| 4 | 14 | 14 |
| 3 | 13 | 13 |
| 2 | 10 | 12 |
| 1 | 9 | 9 |
| 0 | 0 | 8 |

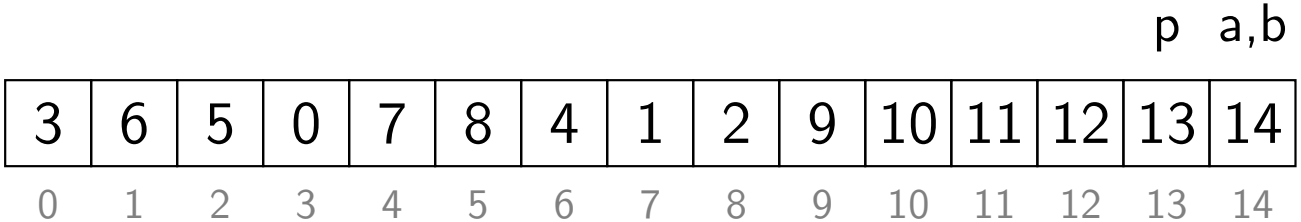
top

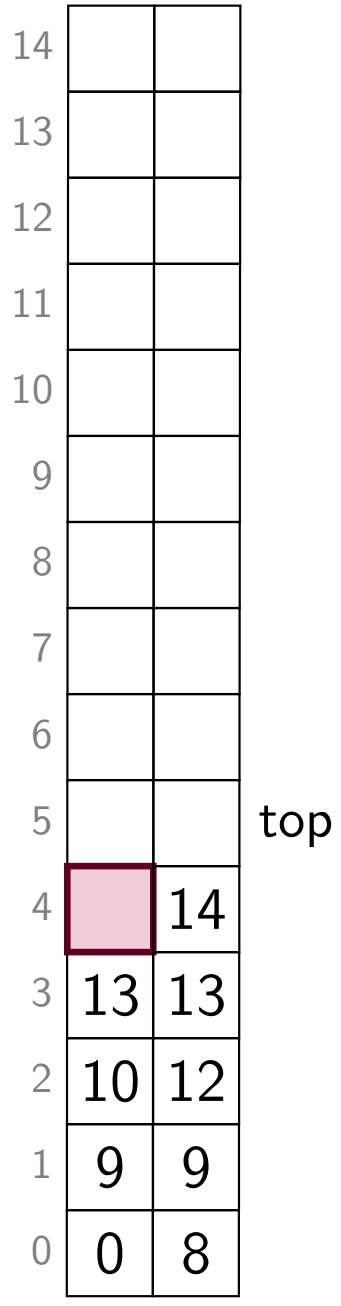
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



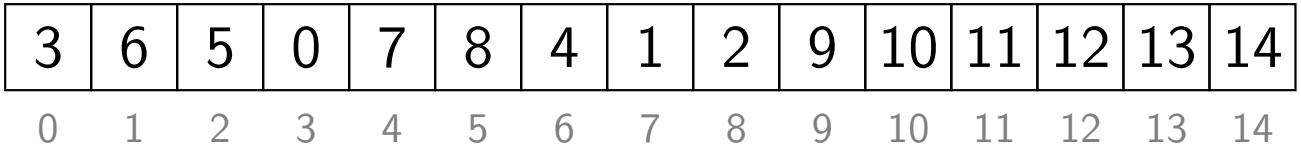



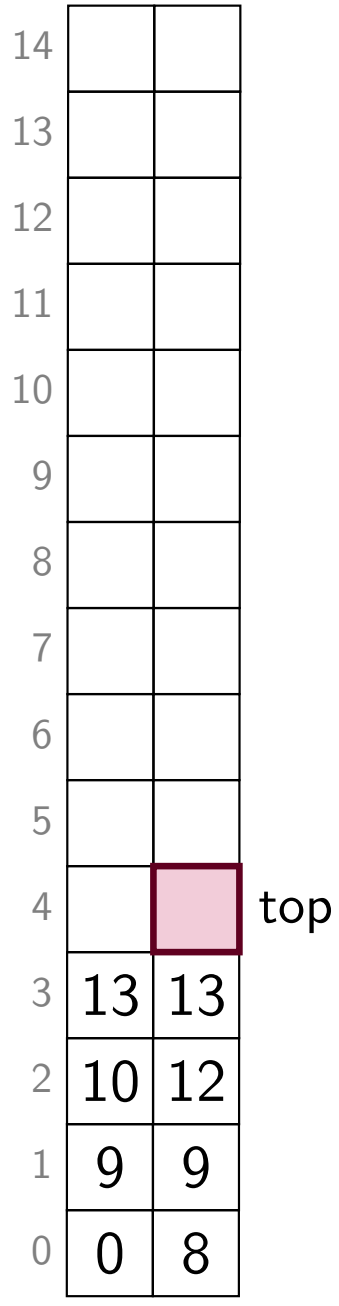
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



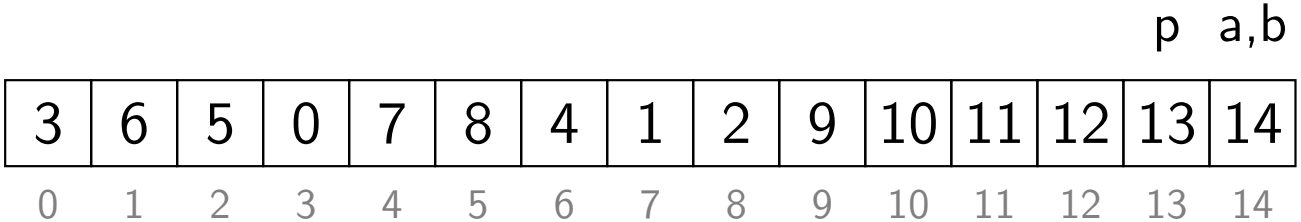



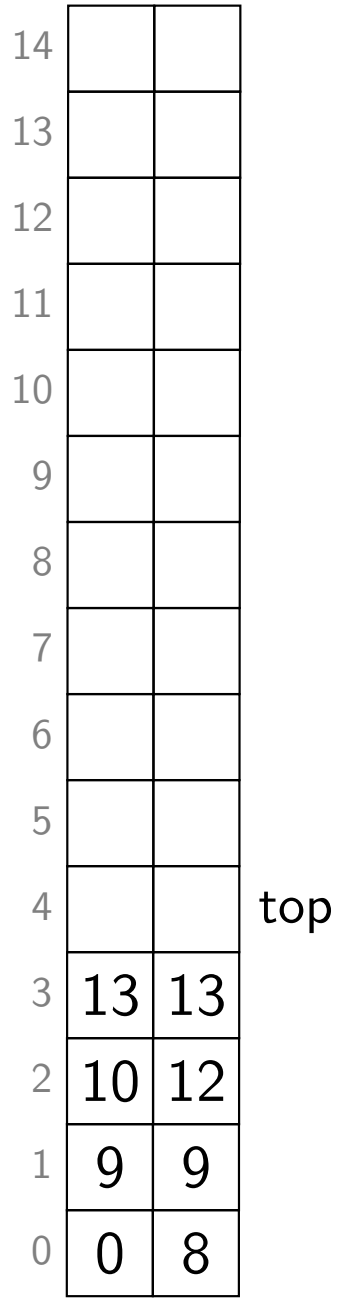
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



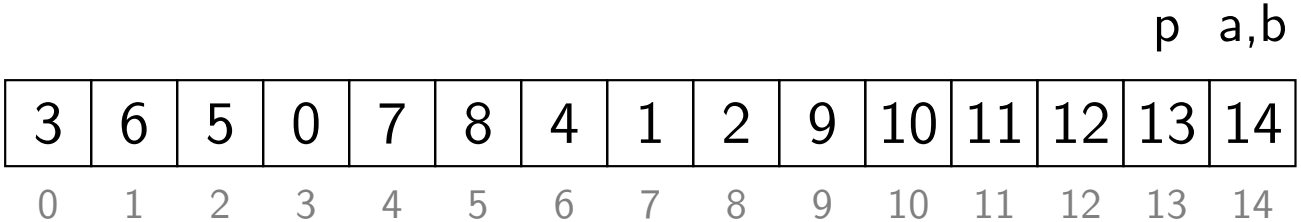



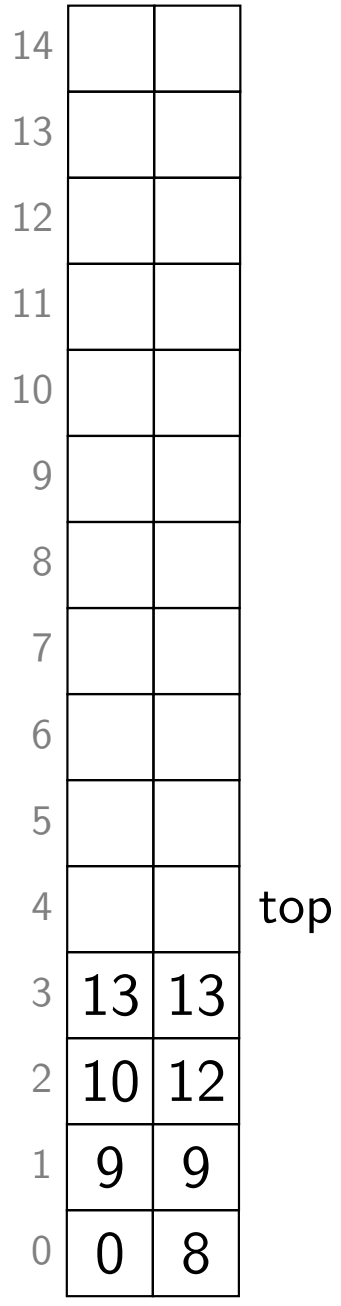
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

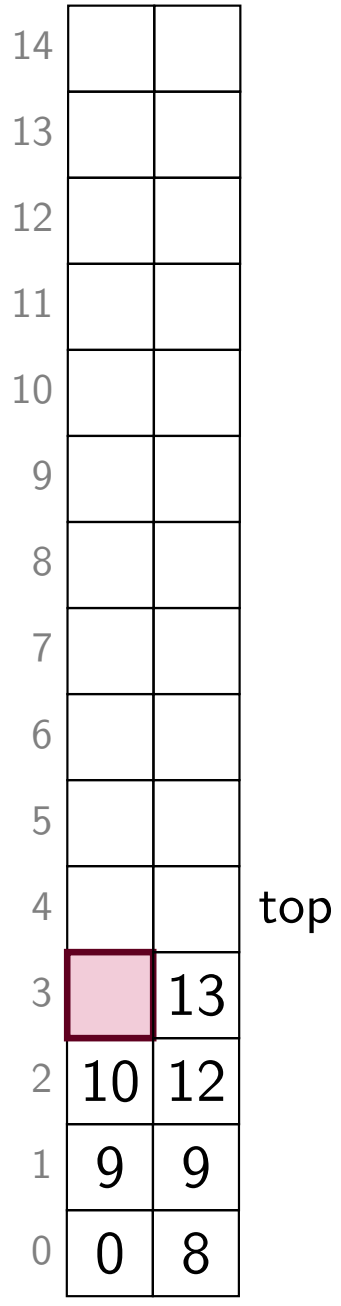


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

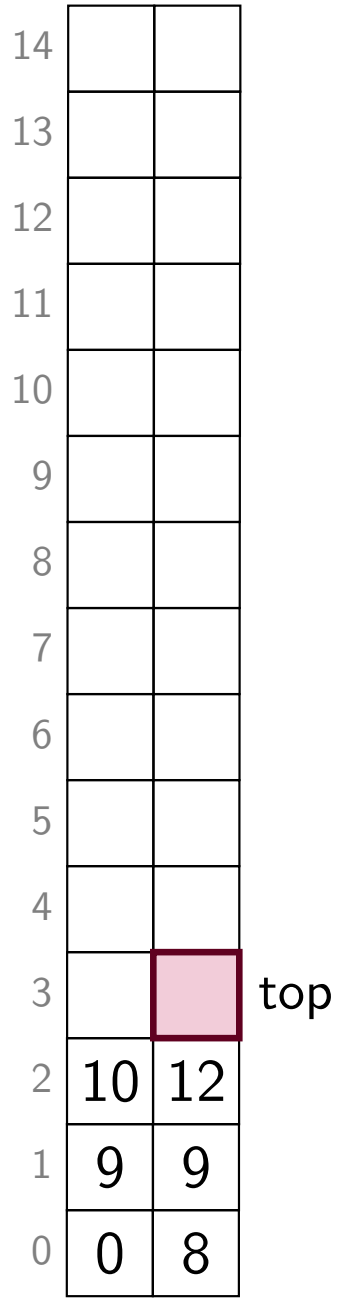
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

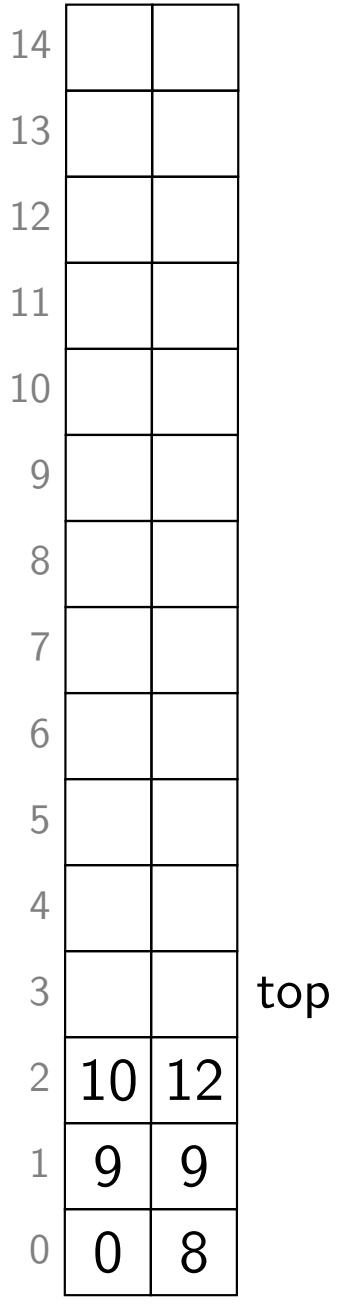
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| | | |
|---|---|---|
| 14 | | |
| 13 | | |
| 12 | | |
| 11 | | |
| 10 | | |
| 9 | | |
| 8 | | |
| 7 | | |
| 6 | | |
| 5 | | |
| 4 | | |
| 3 | | | top |
| 2 | 10 | 12 | |
| 1 | 9 | 9 | |
| 0 | 0 | 8 | |

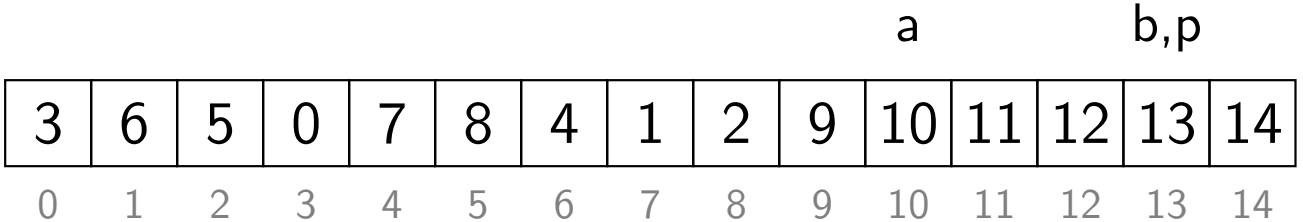
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

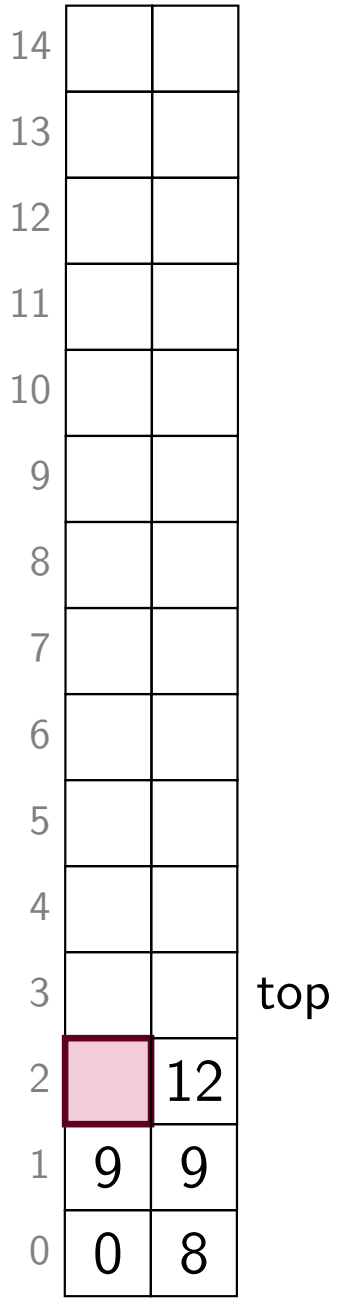
a,b,p

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

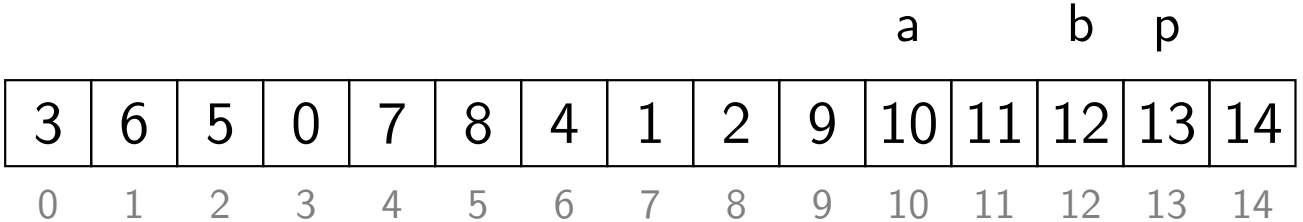


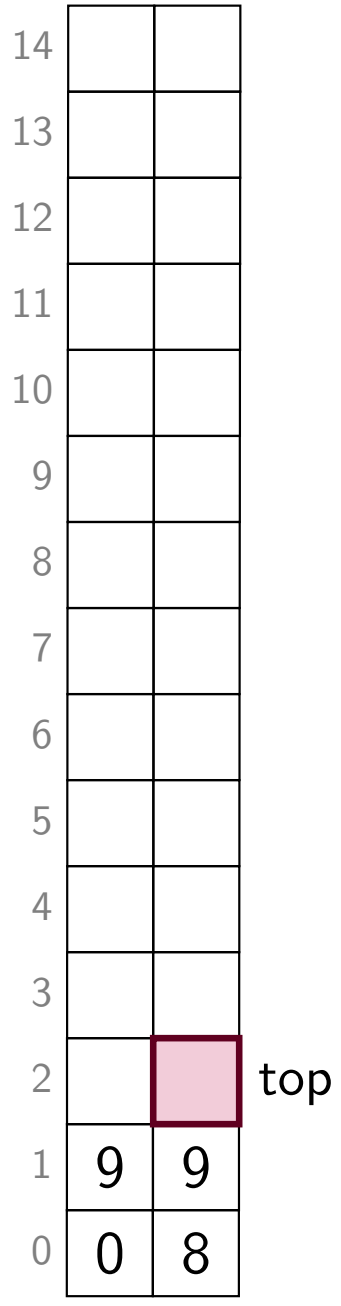
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



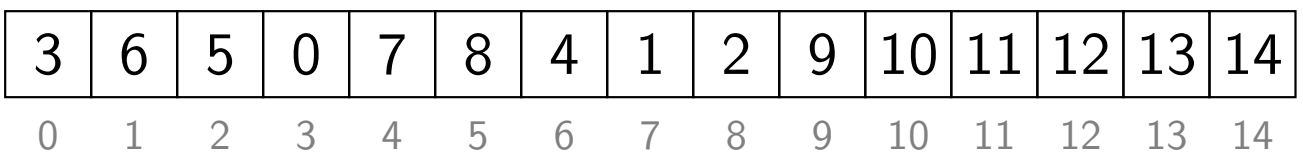



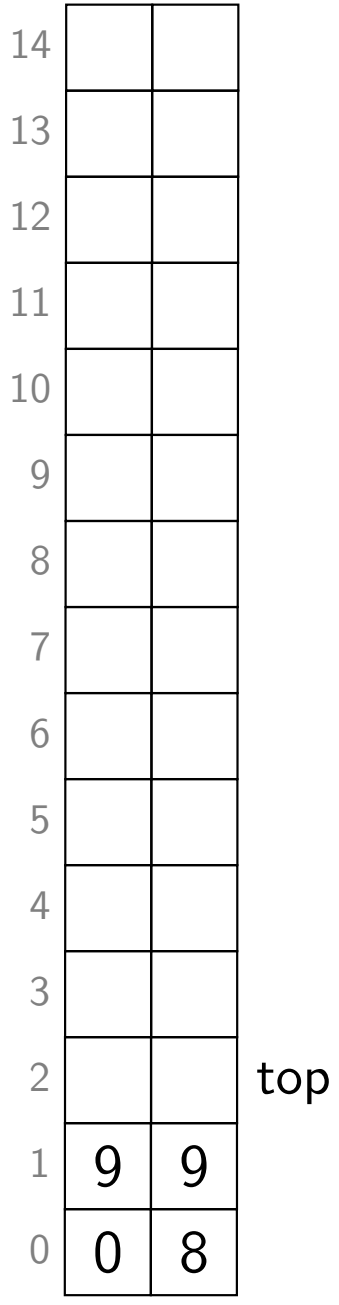
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



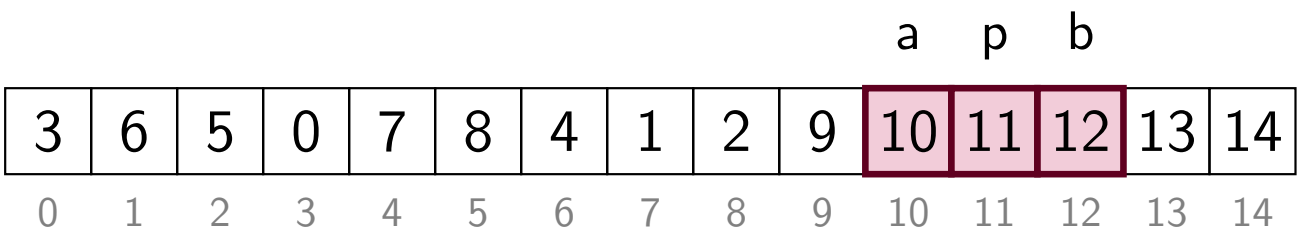



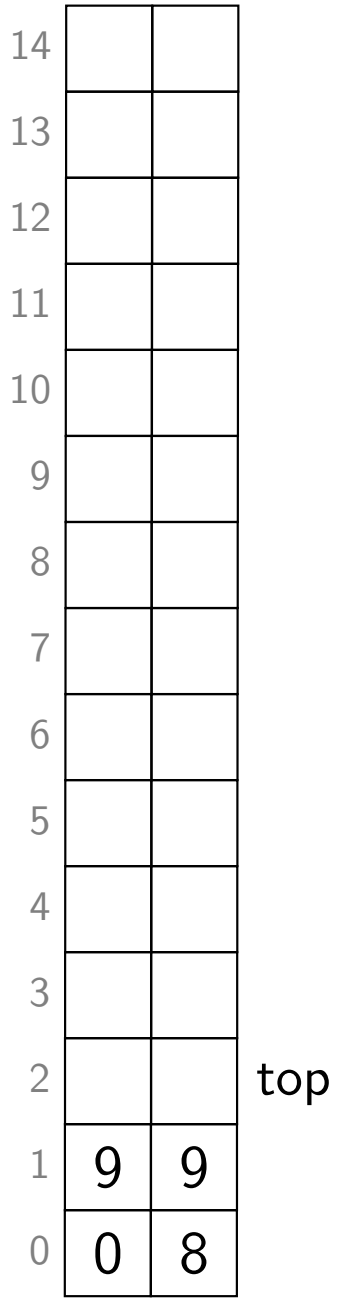
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



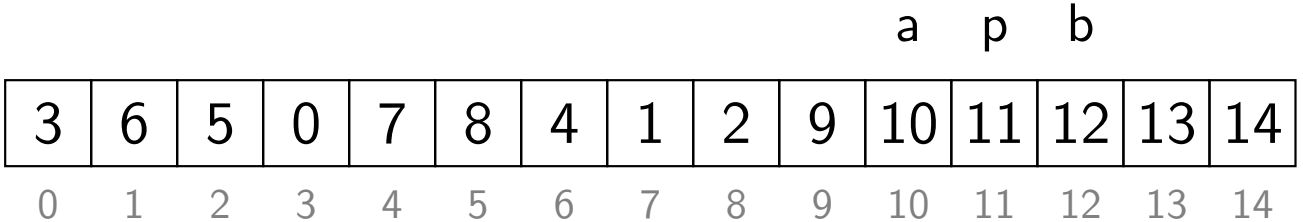



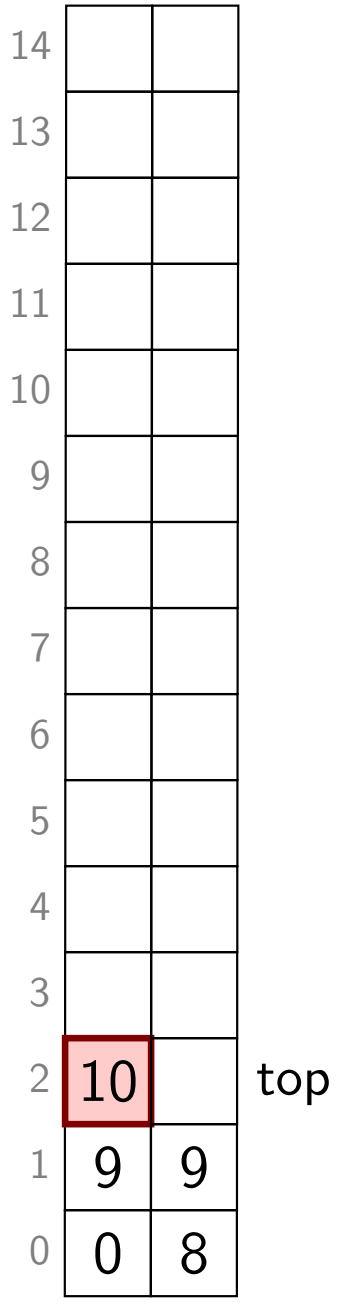
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



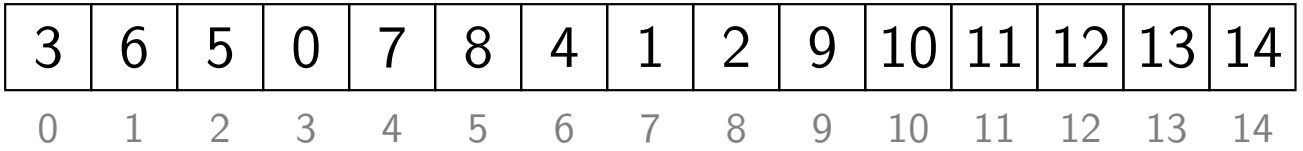



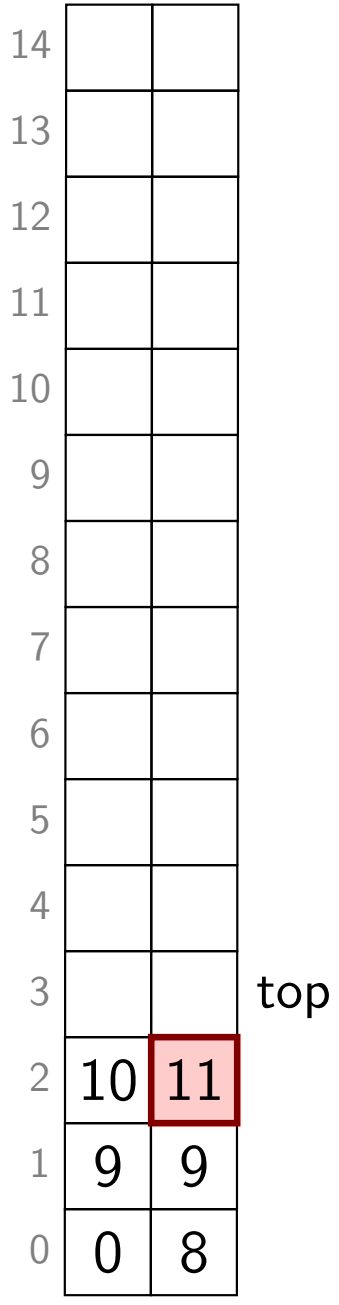
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



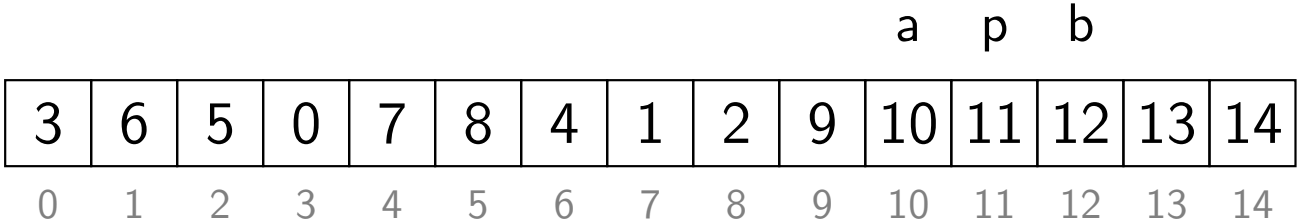



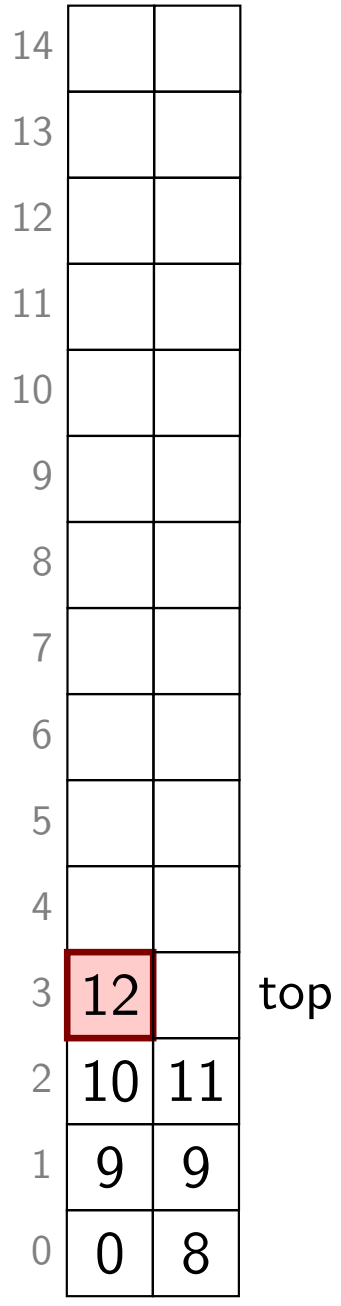
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



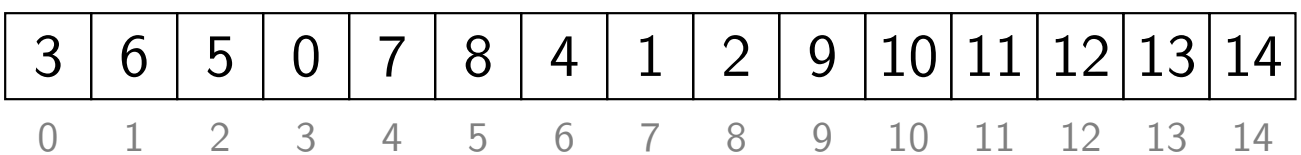



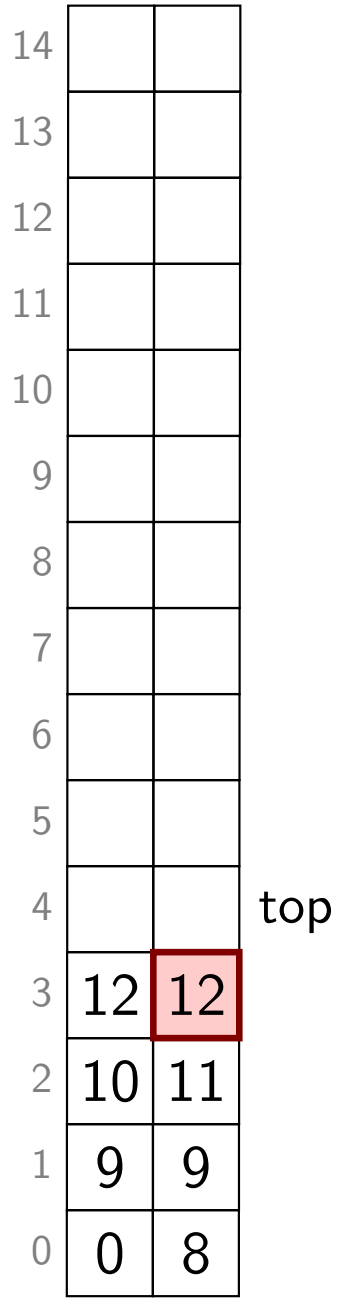
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



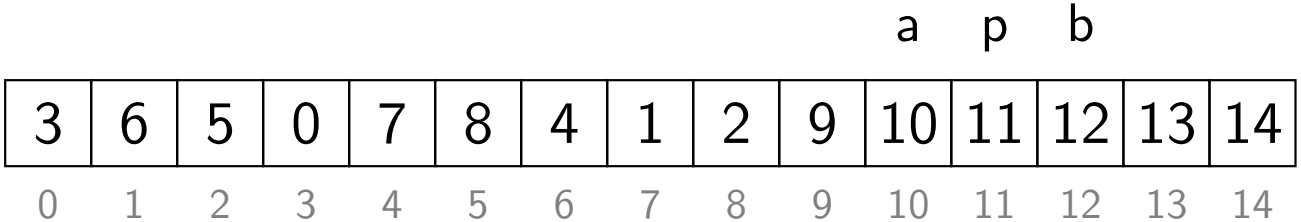



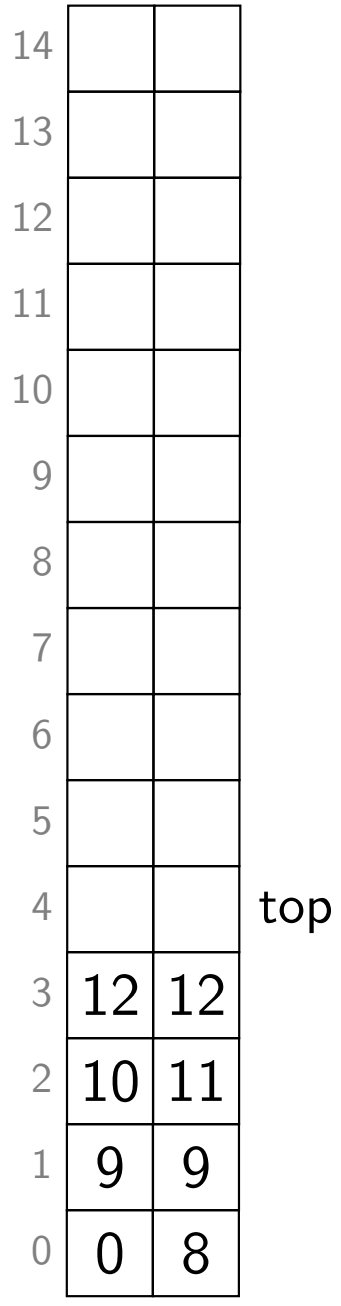
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



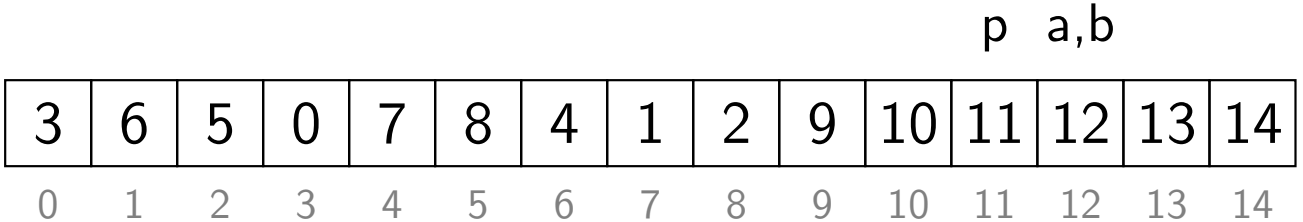



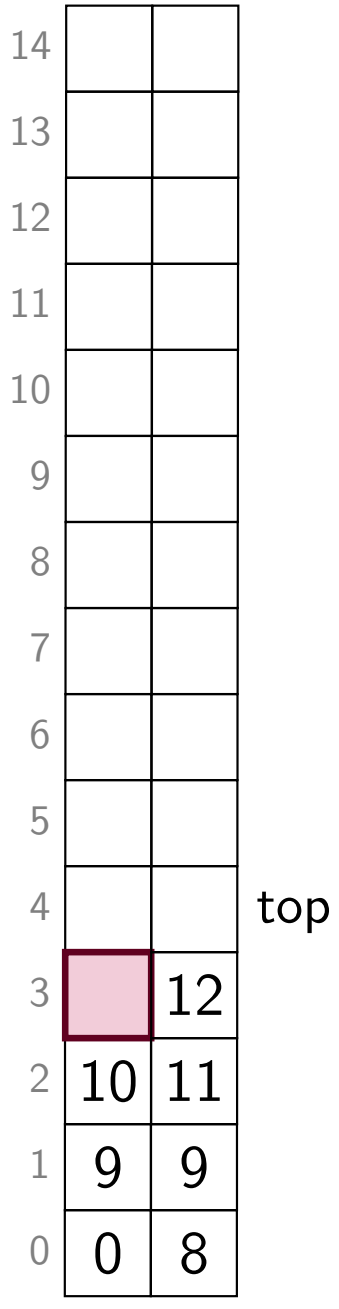
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



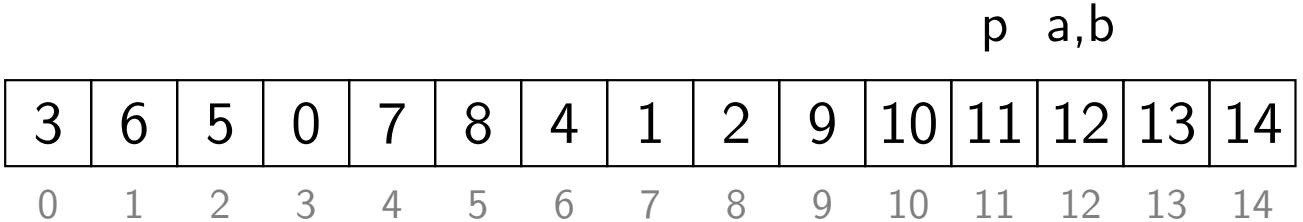



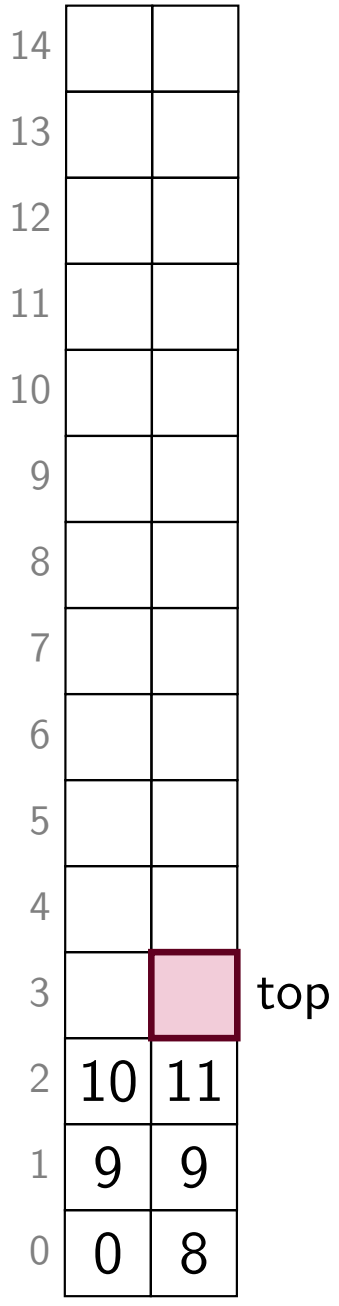
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



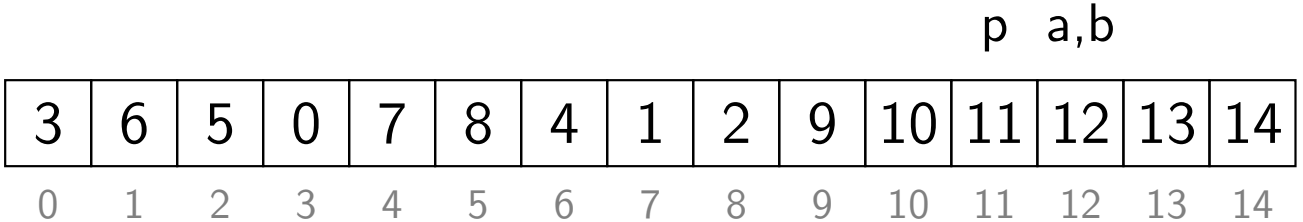
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

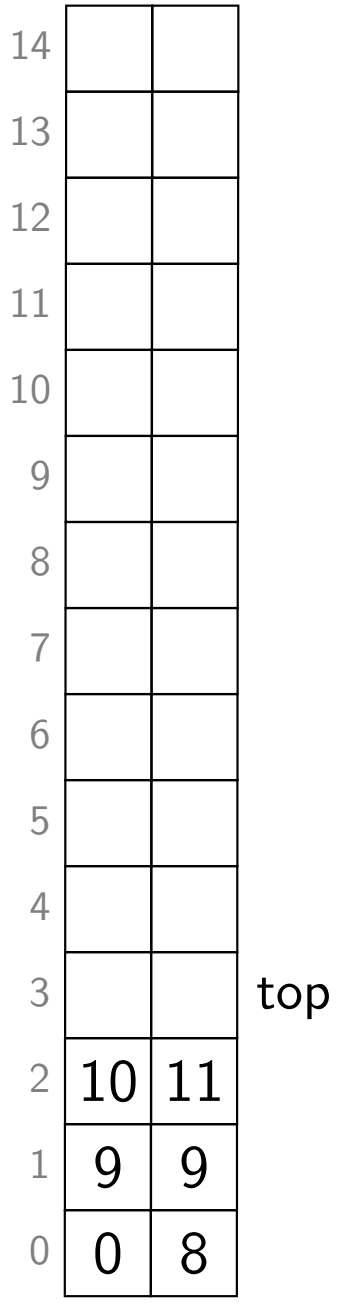
p a,b

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

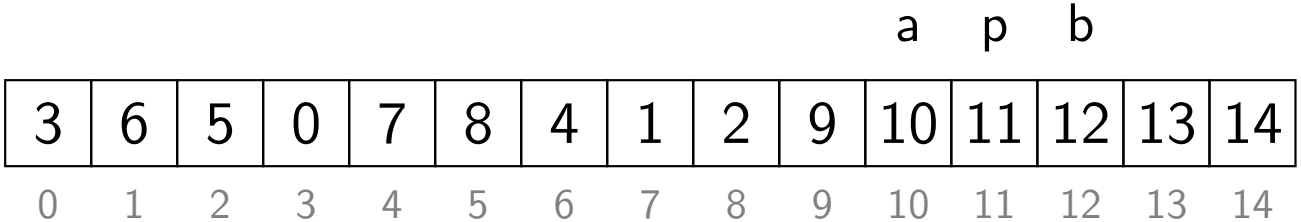
| | | |
|---|---|---|
| 14 | | |
| 13 | | |
| 12 | | |
| 11 | | |
| 10 | | |
| 9 | | |
| 8 | | |
| 7 | | |
| 6 | | |
| 5 | | |
| 4 | | |
| 3 | | | top |
| 2 | 10 | 11 | |
| 1 | 9 | 9 | |
| 0 | 0 | 8 | |

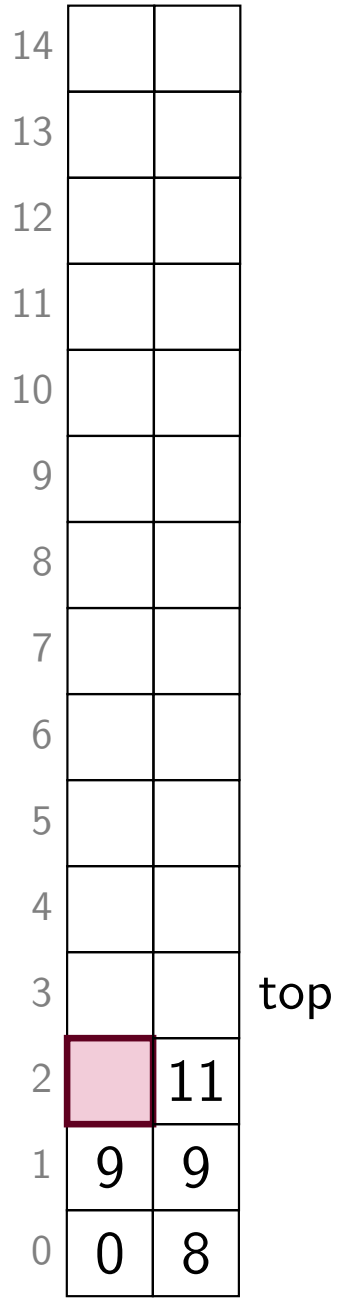
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



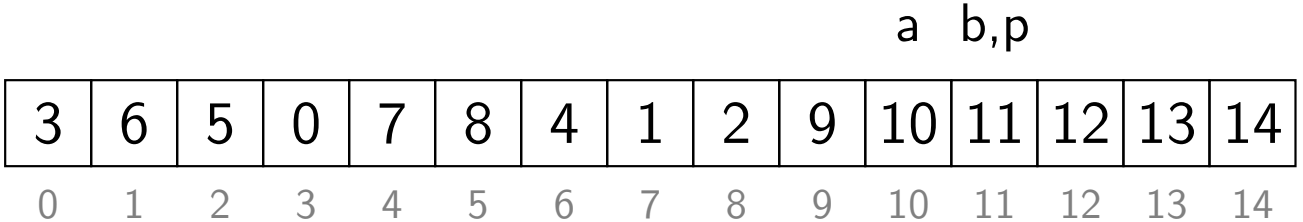



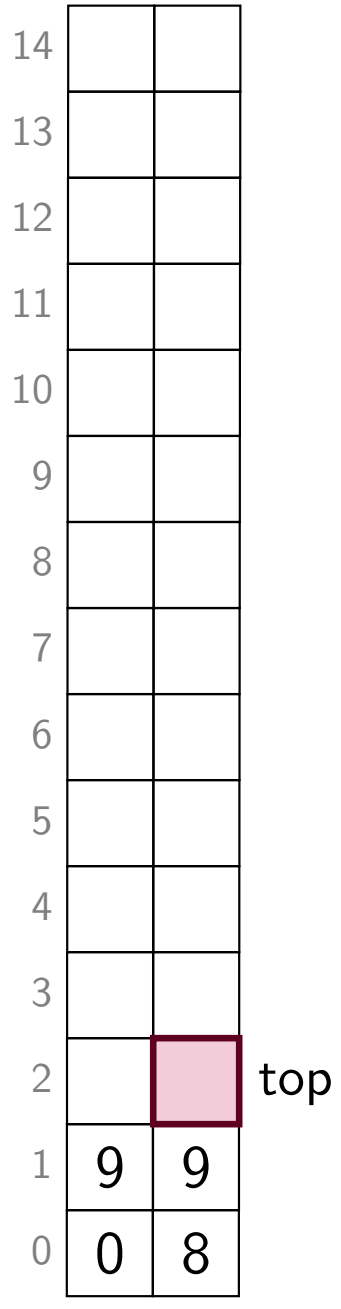
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



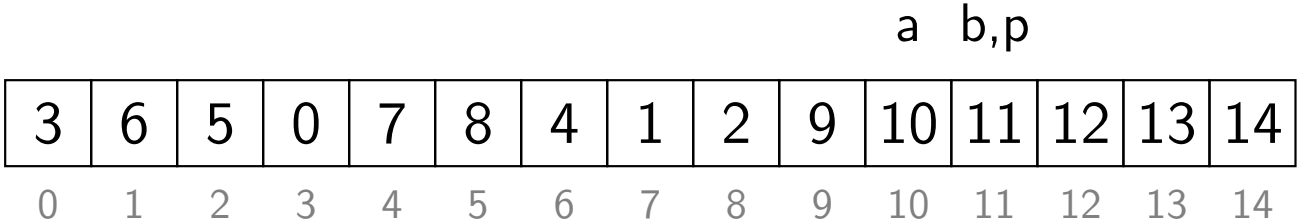



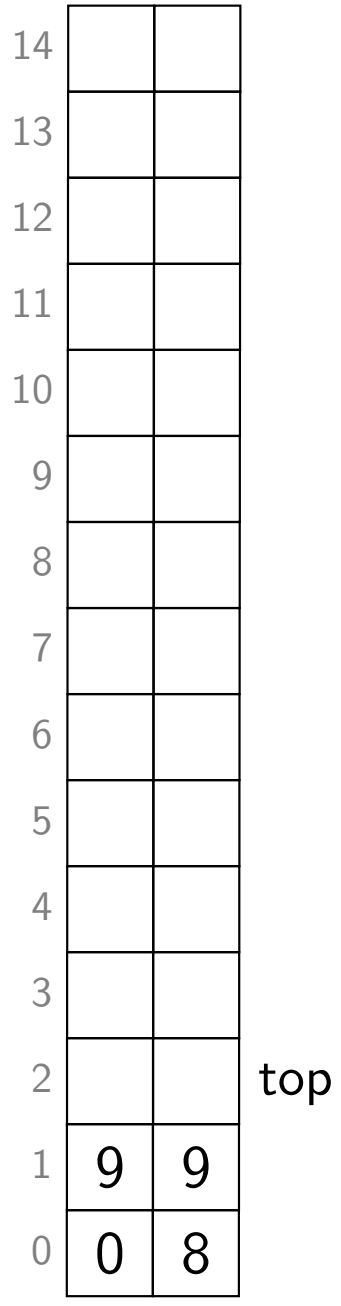
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



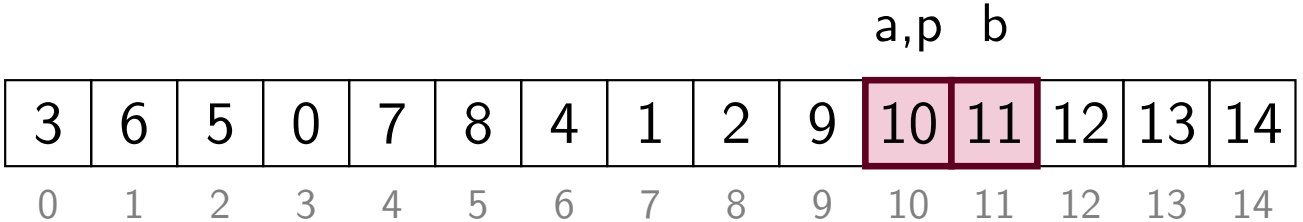



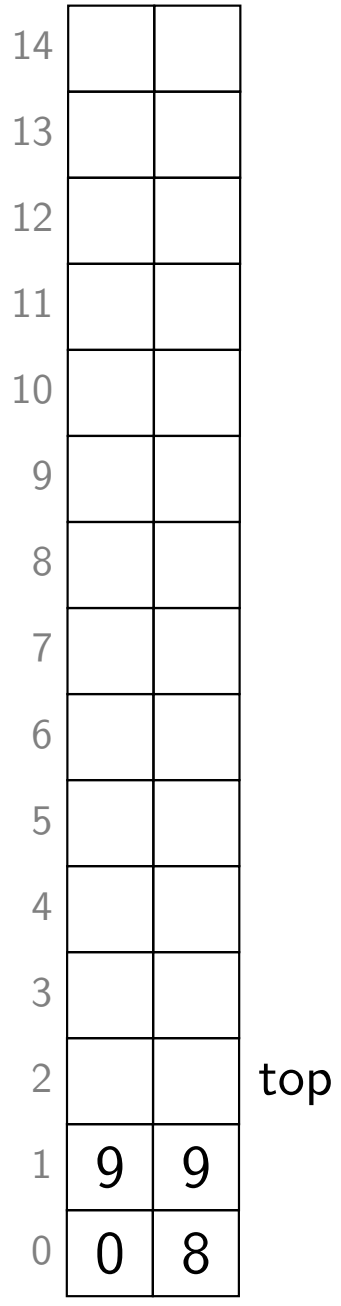
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

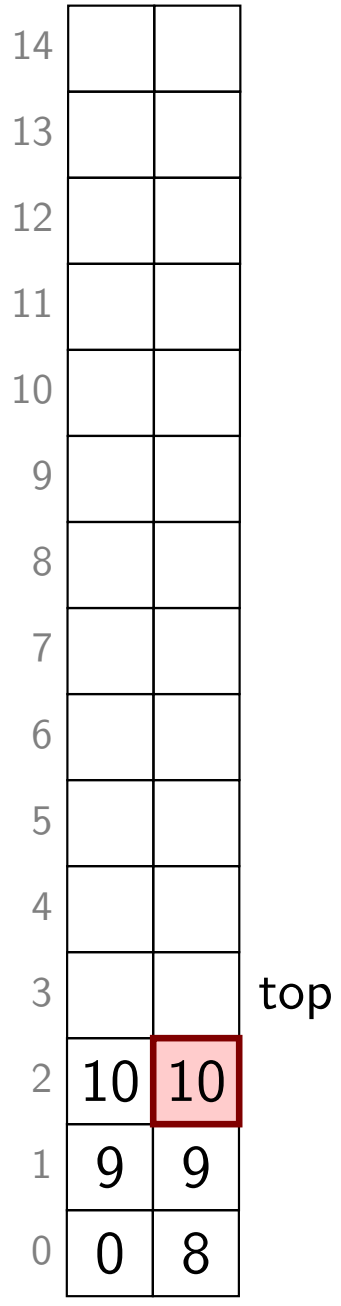
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| | | | |
|---|---|---|---|
| 14 | | | |
| 13 | | | |
| 12 | | | |
| 11 | | | |
| 10 | | | |
| 9 | | | |
| 8 | | | |
| 7 | | | |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | | | |
| 2 | 10 | | top |
| 1 | 9 | 9 | |
| 0 | 0 | 8 | |

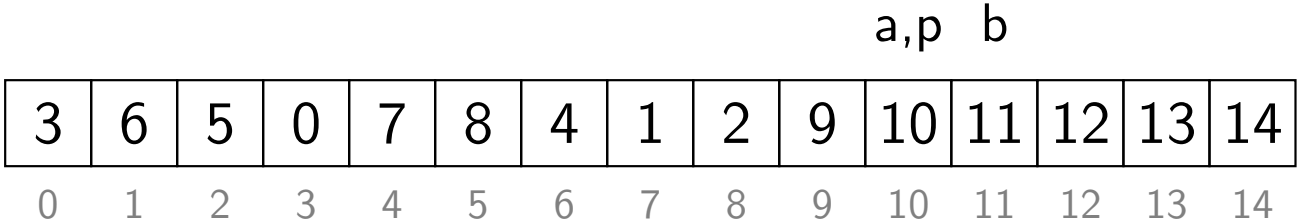
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

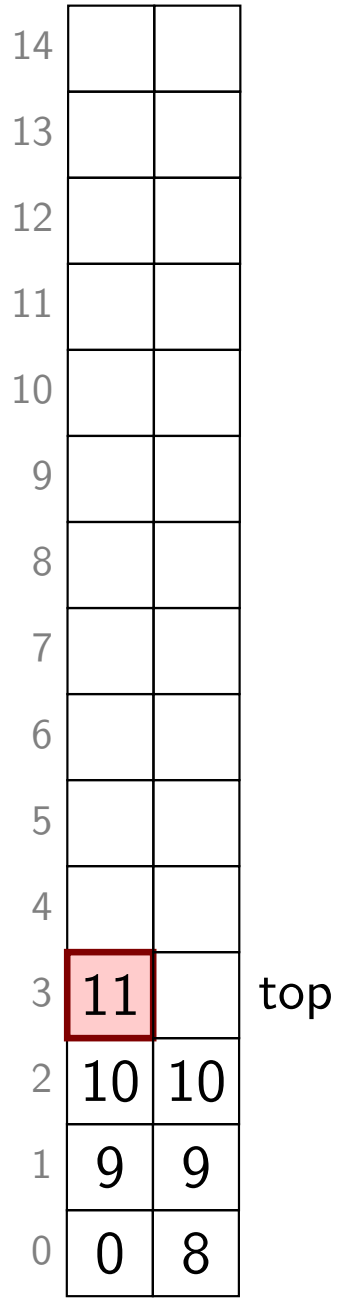
a,p b

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

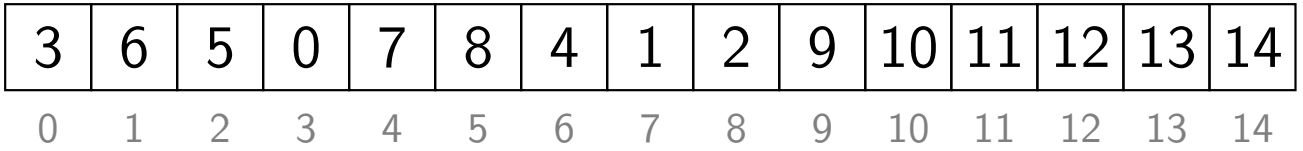


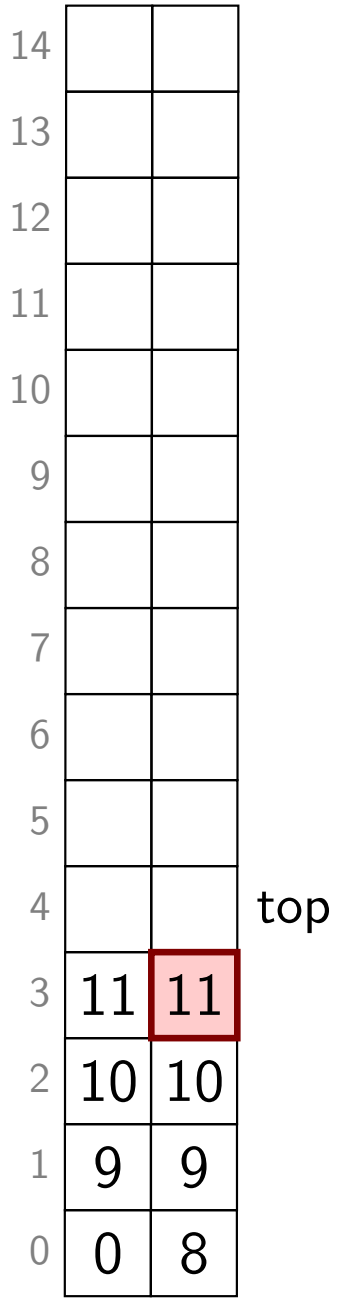
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



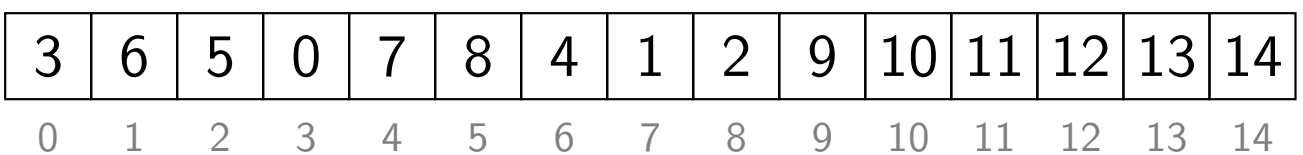



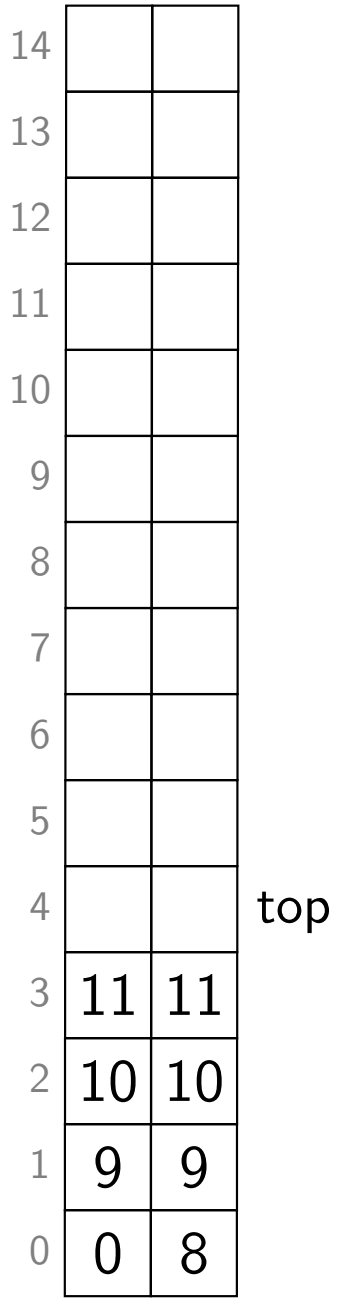
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



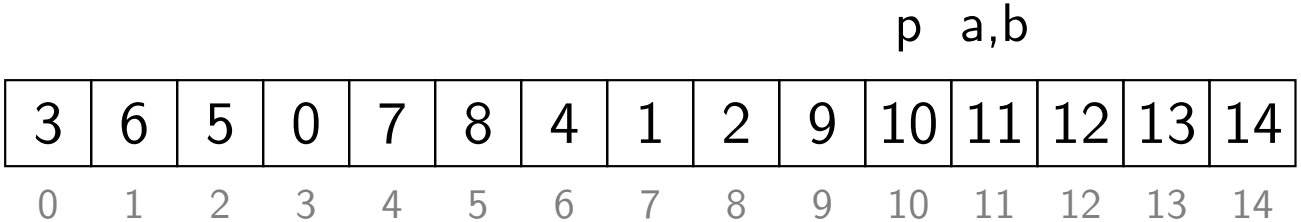



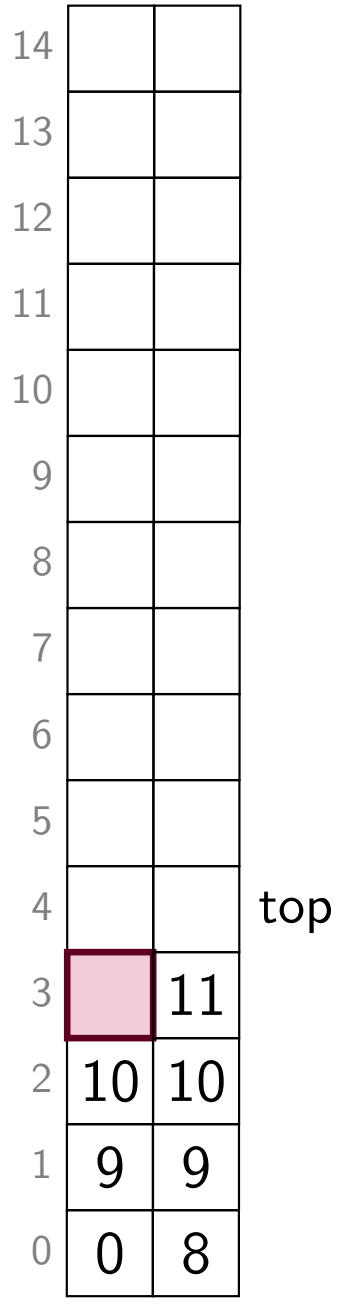
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



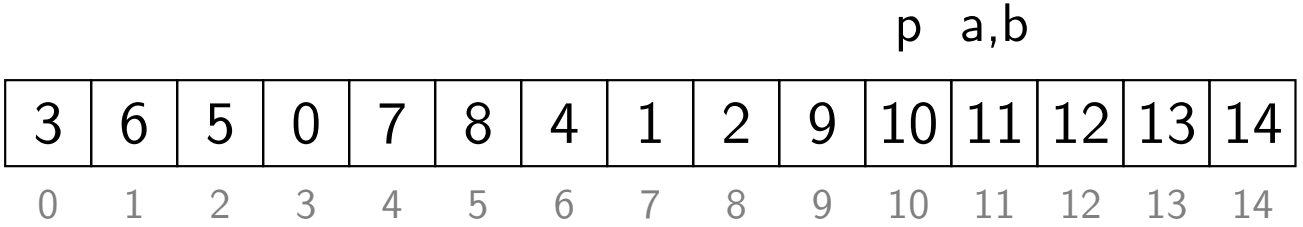



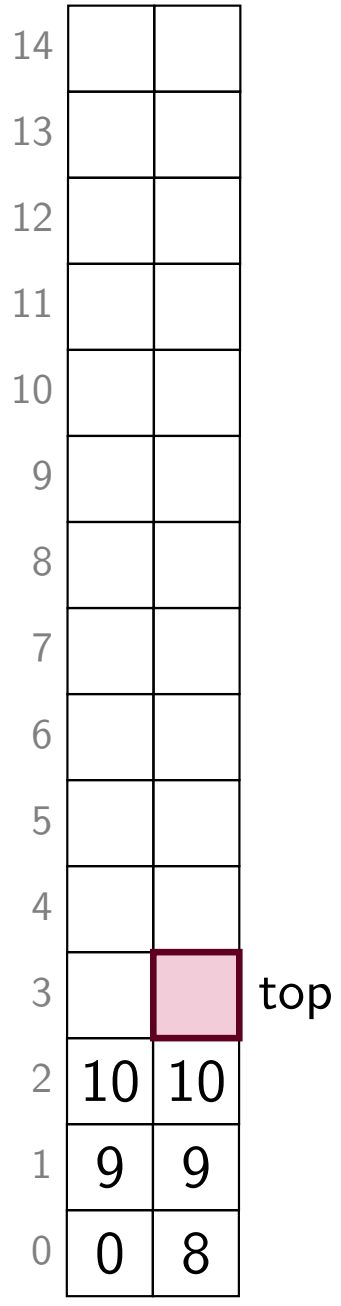
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



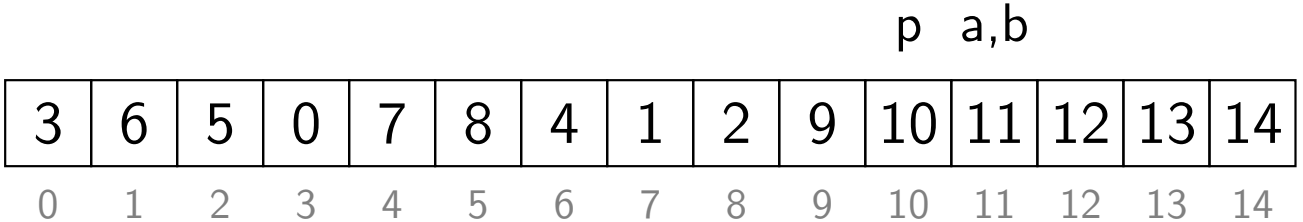



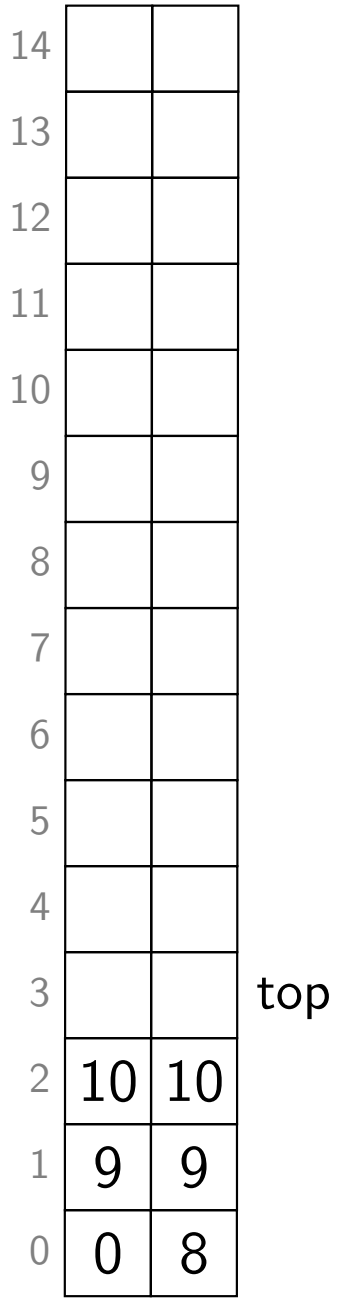
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



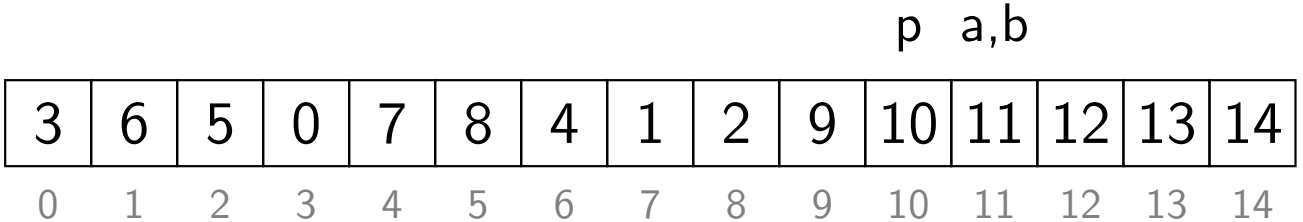



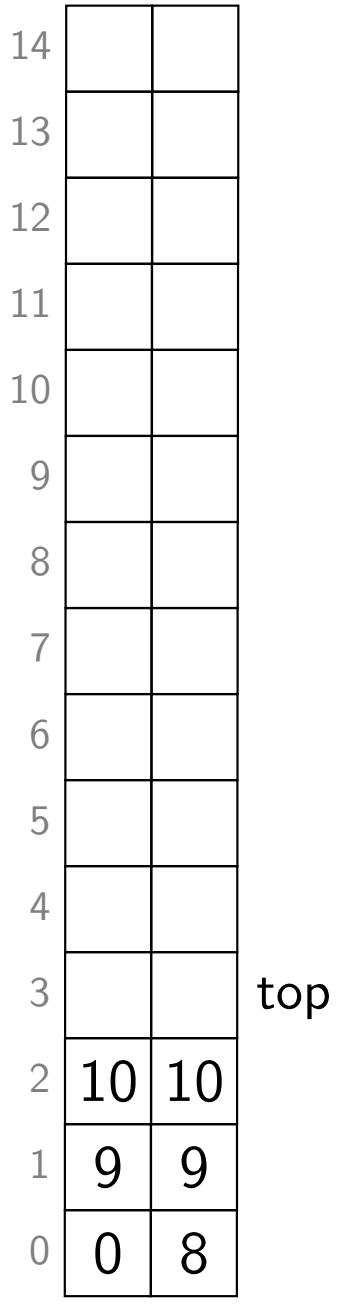
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

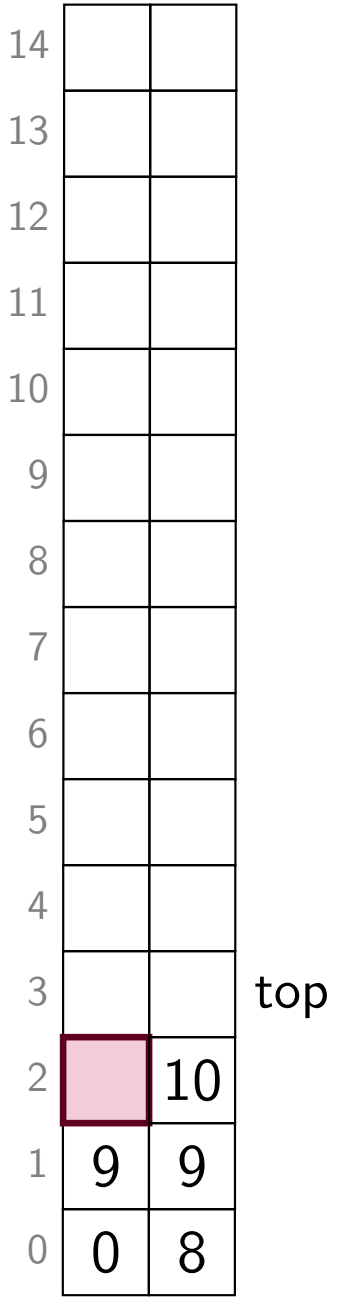


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

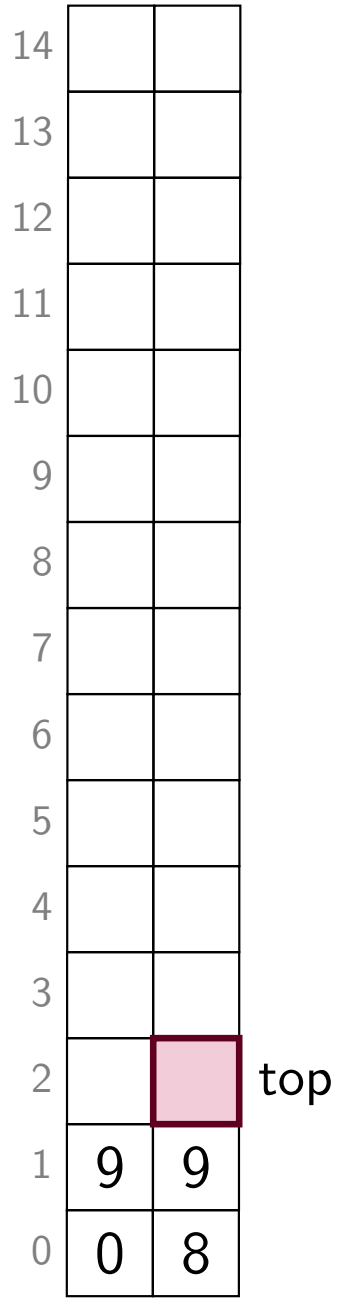
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

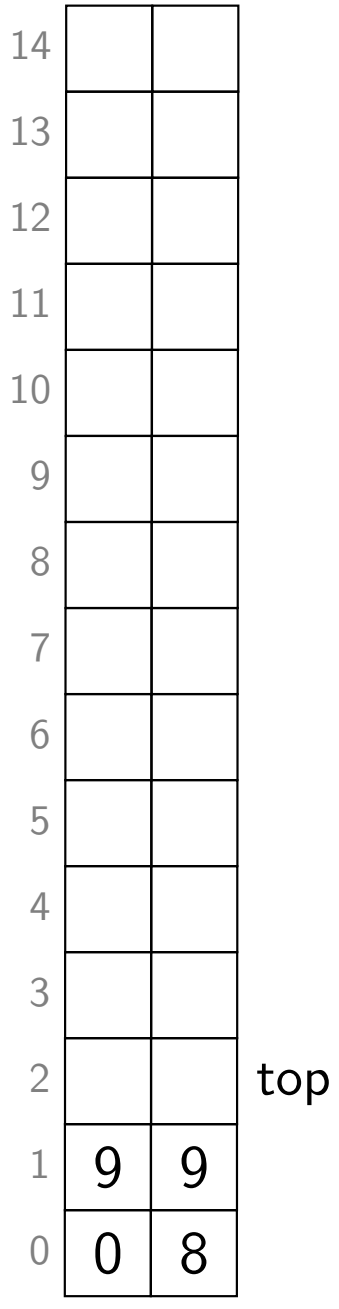
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

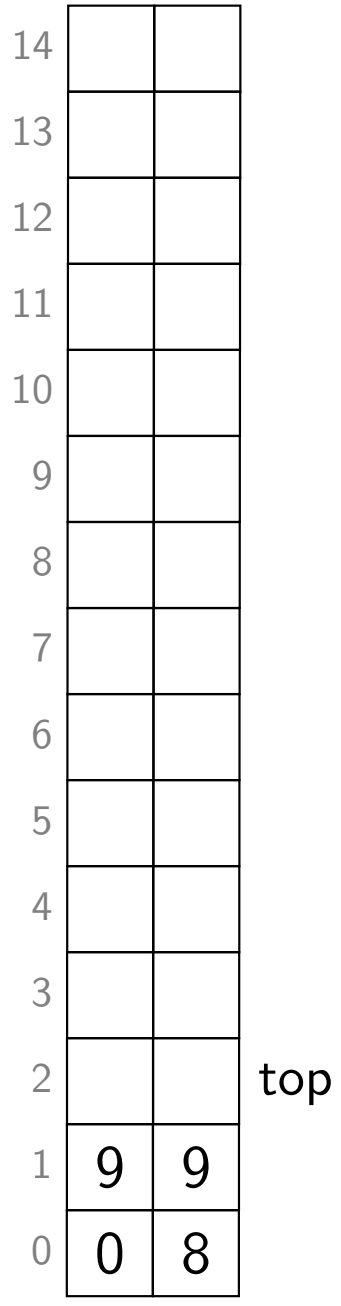
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



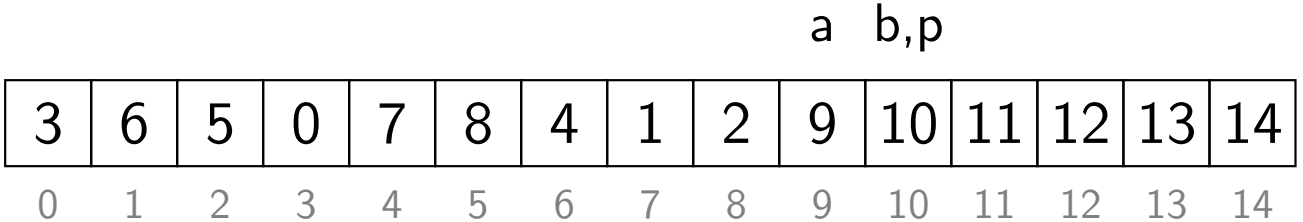
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

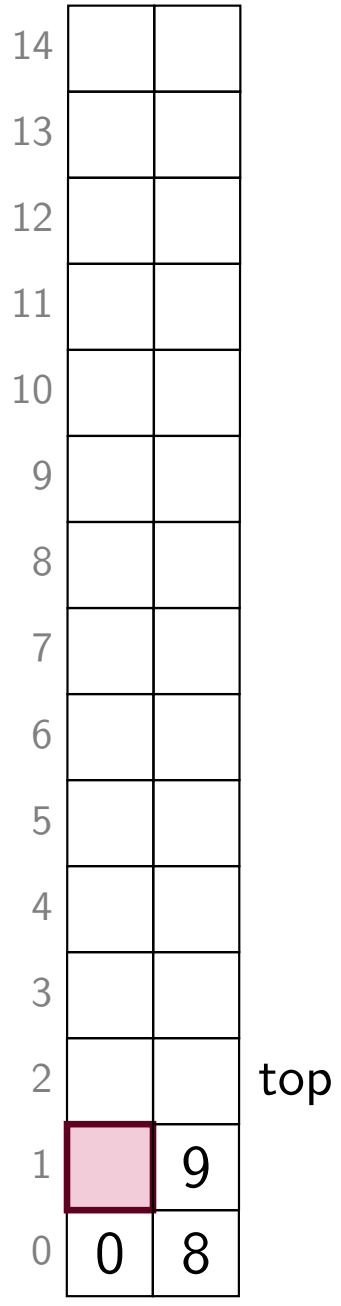
a,b,p

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

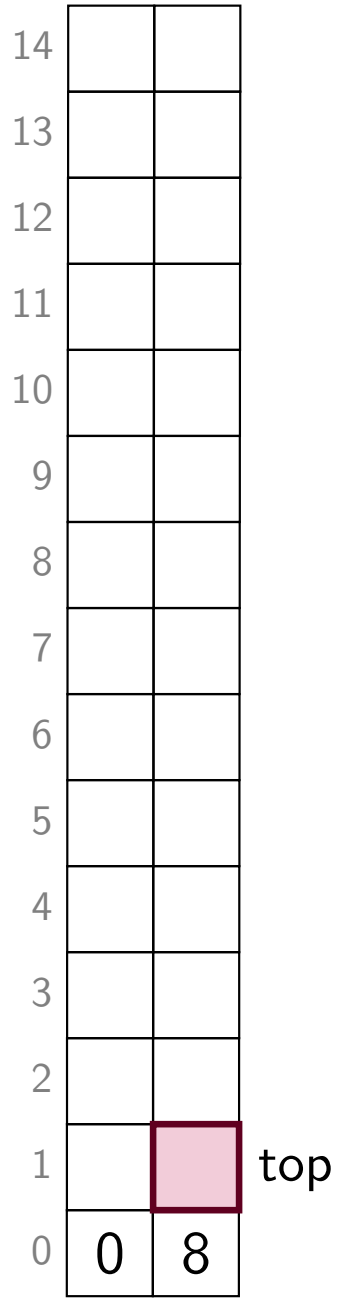


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b p

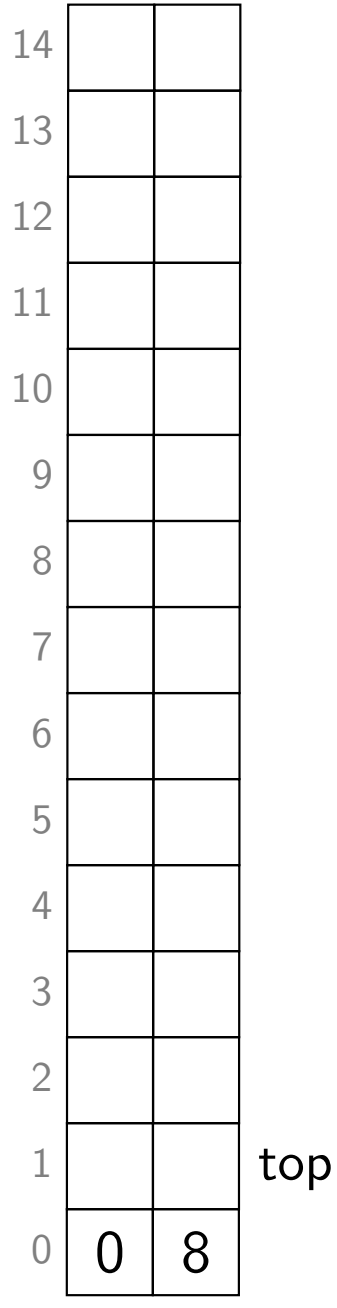
| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



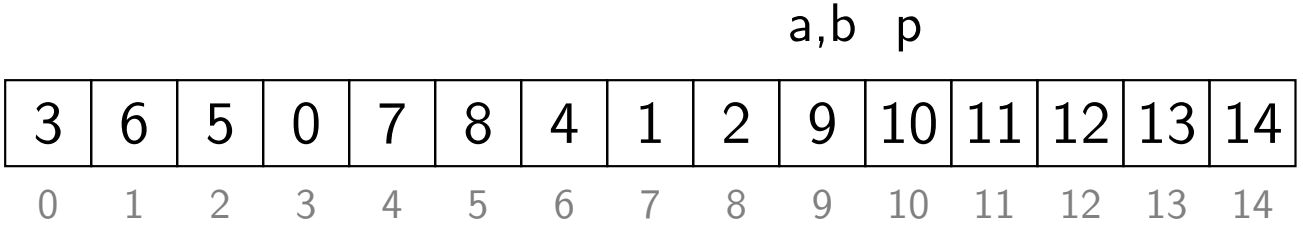
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

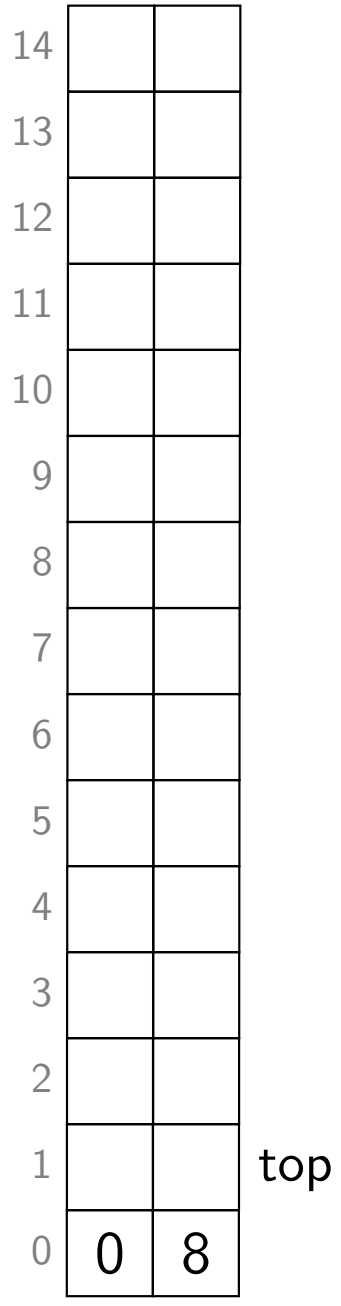
a,b p

| 3 | 6 | 5 | 0 | 7 | 8 | 4 | 1 | 2 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

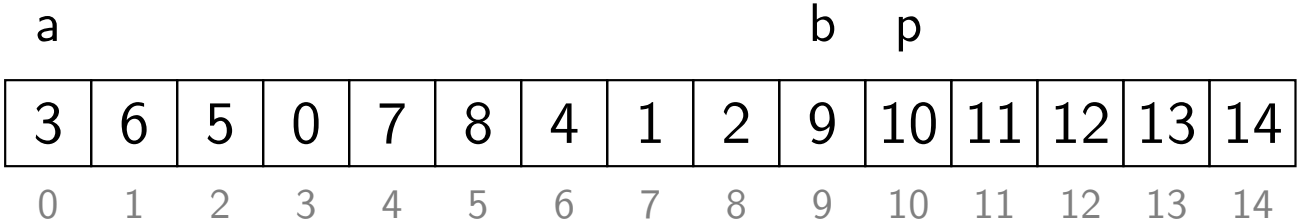


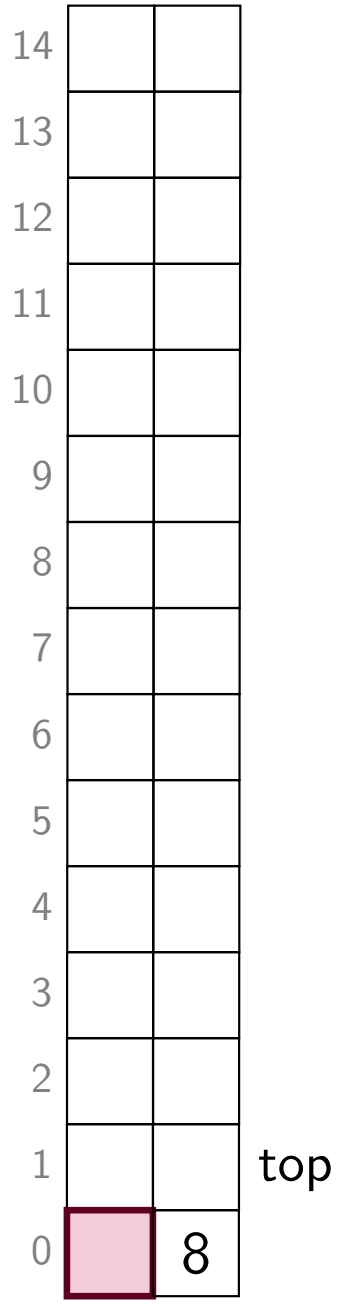
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



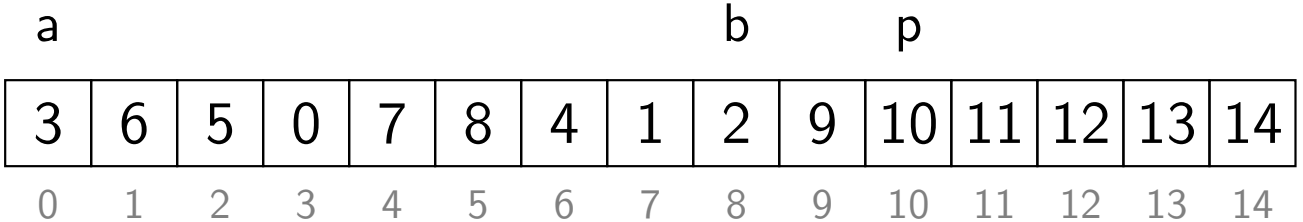



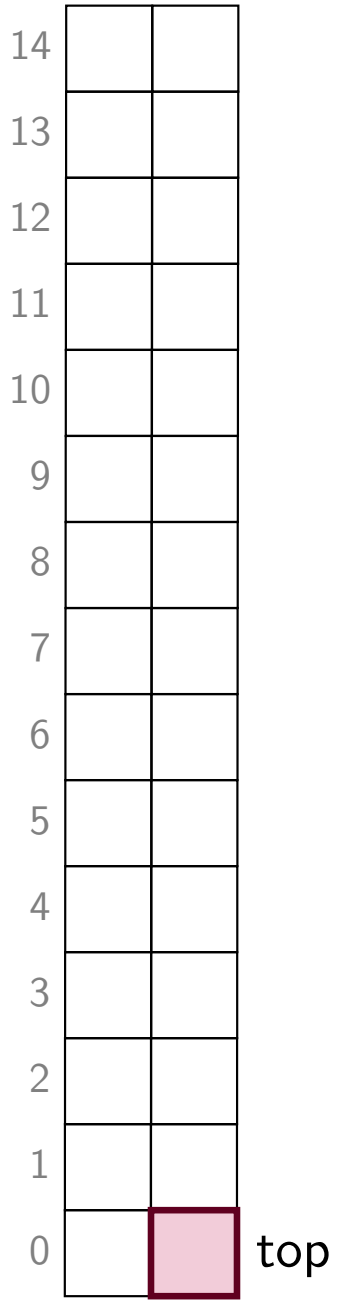
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



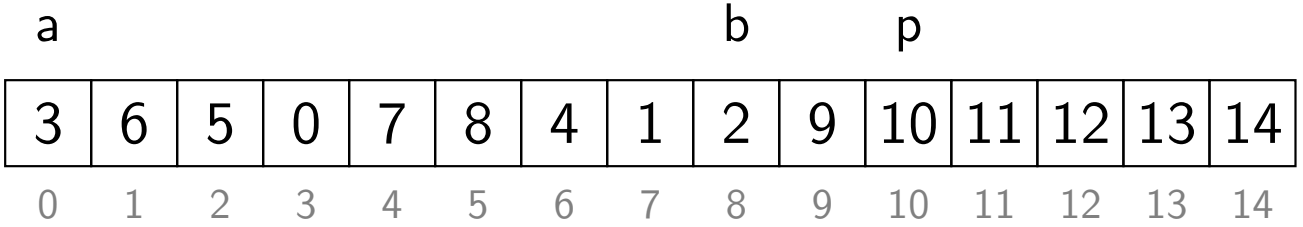



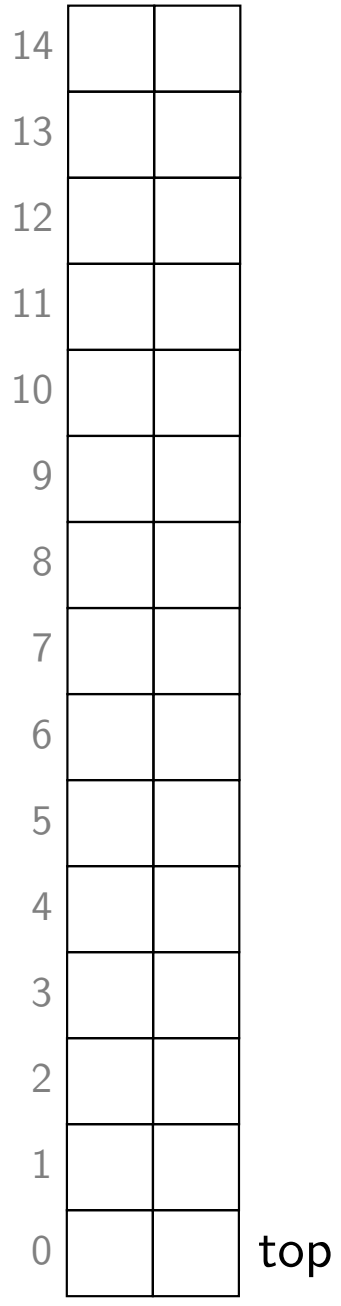
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



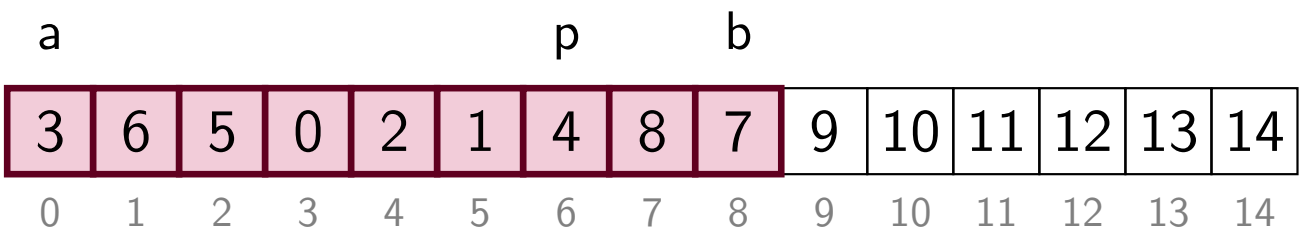



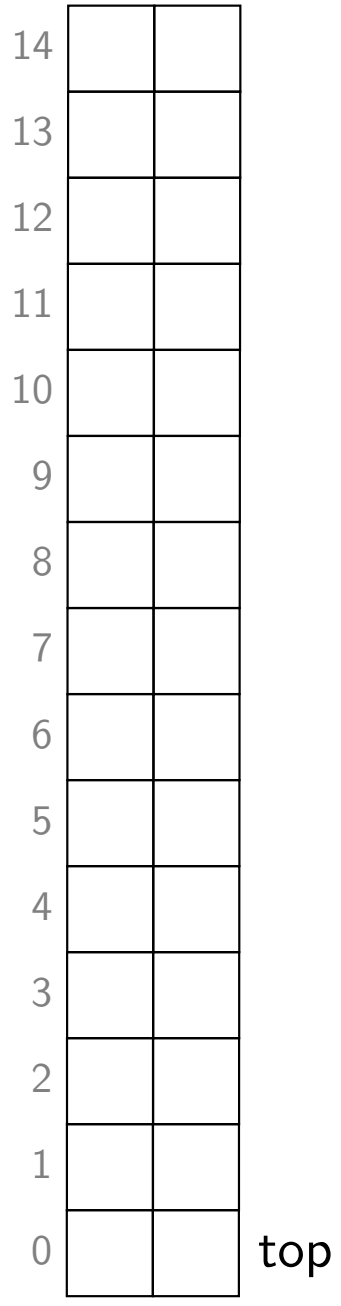
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



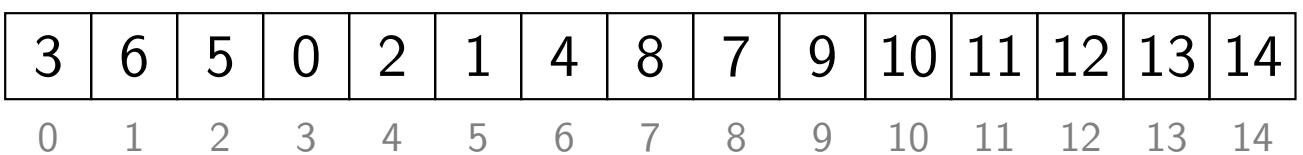



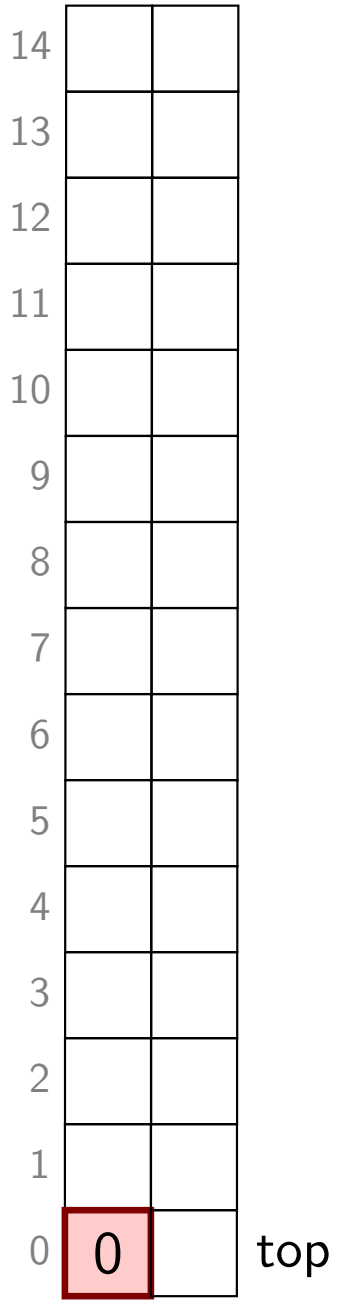
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



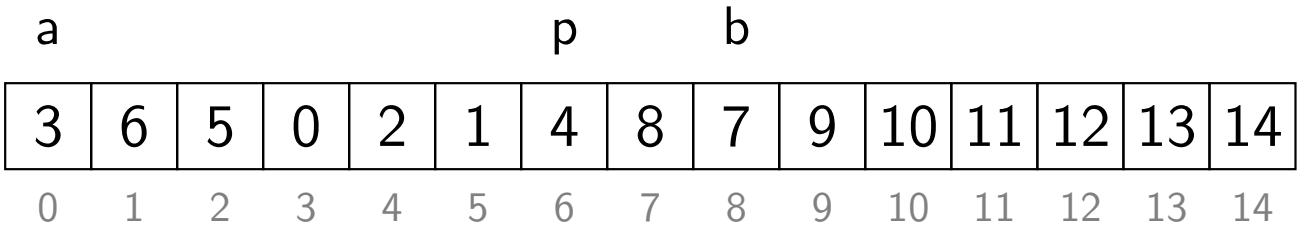



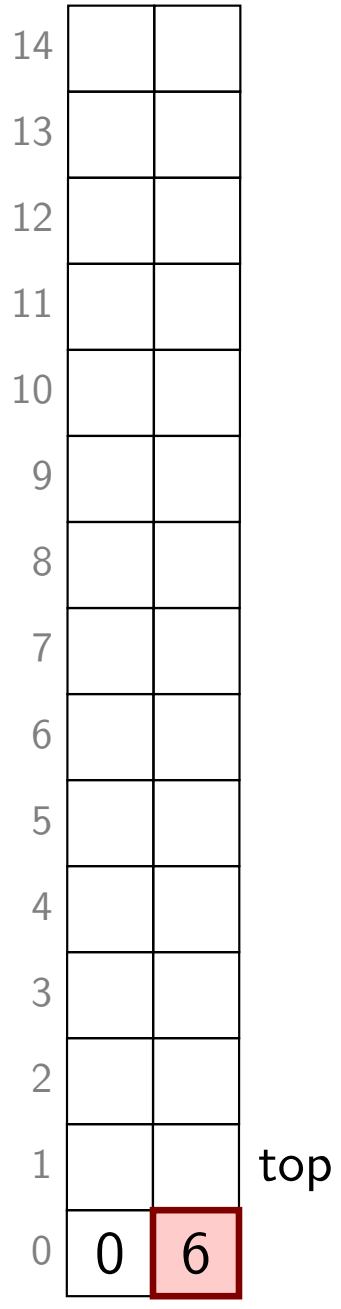
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



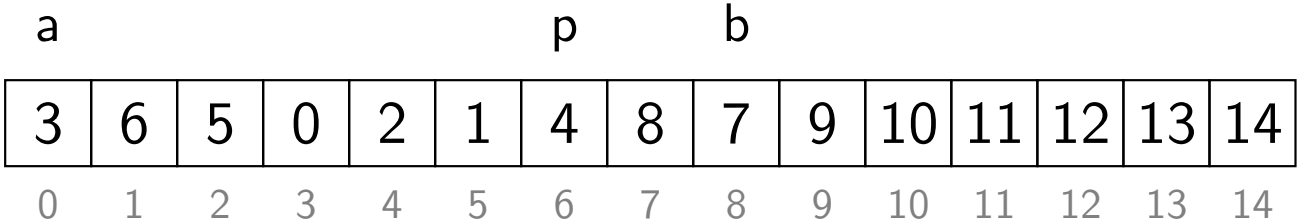



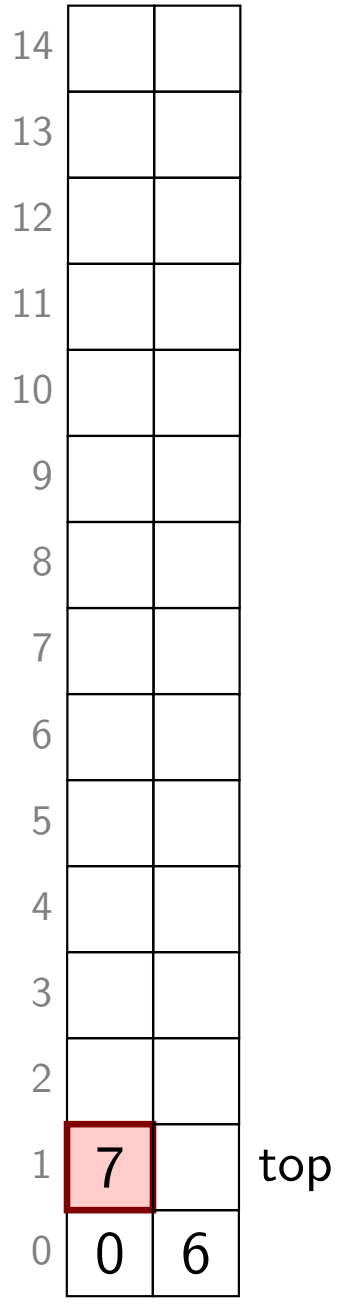
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



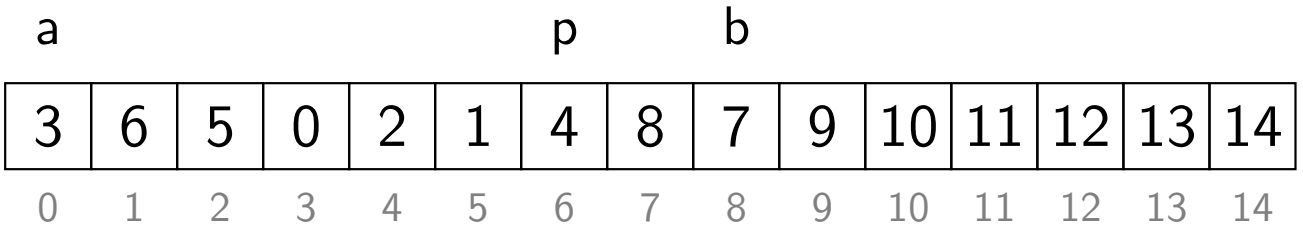



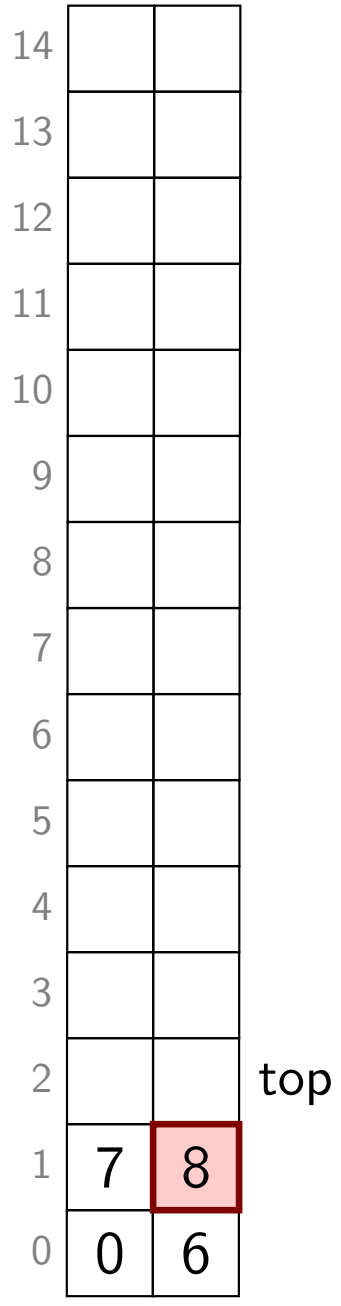
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



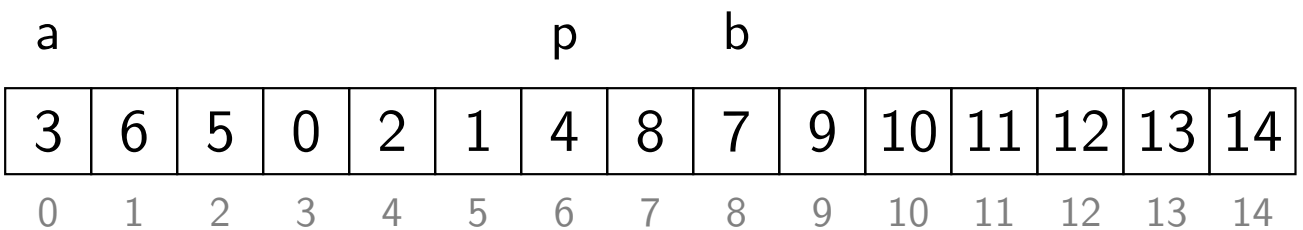



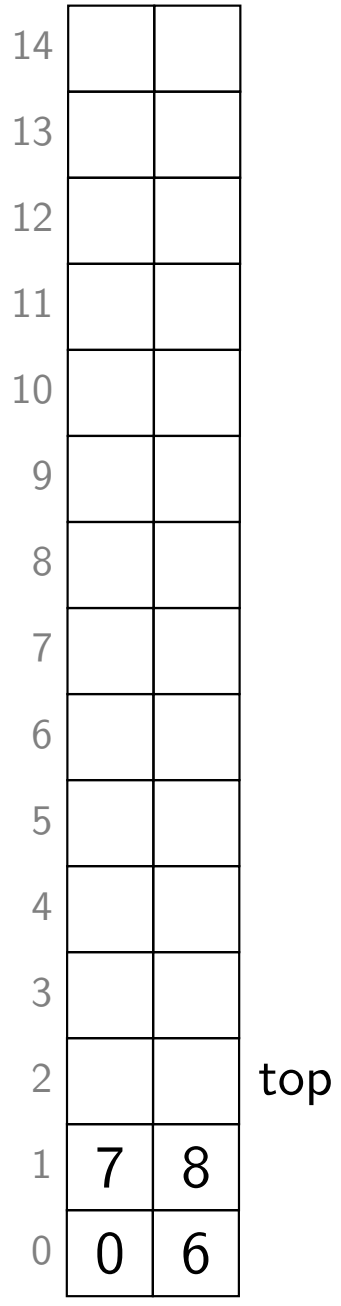
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

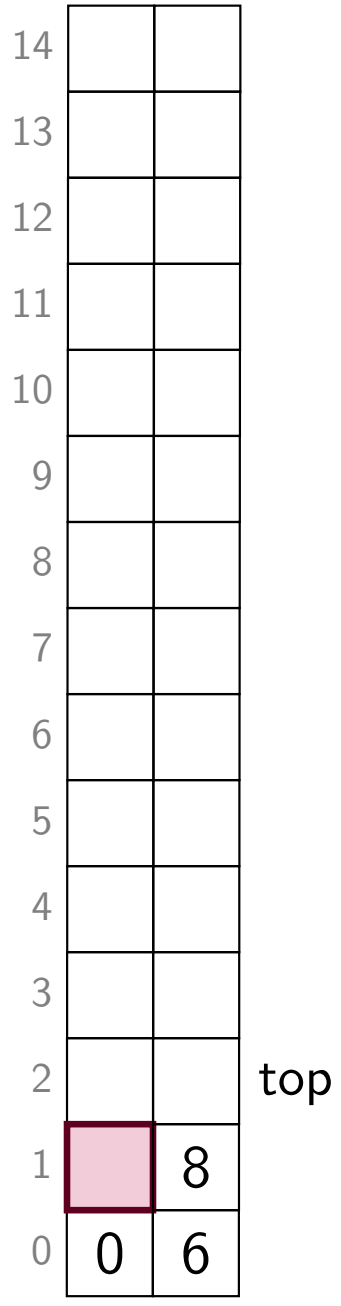


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a b

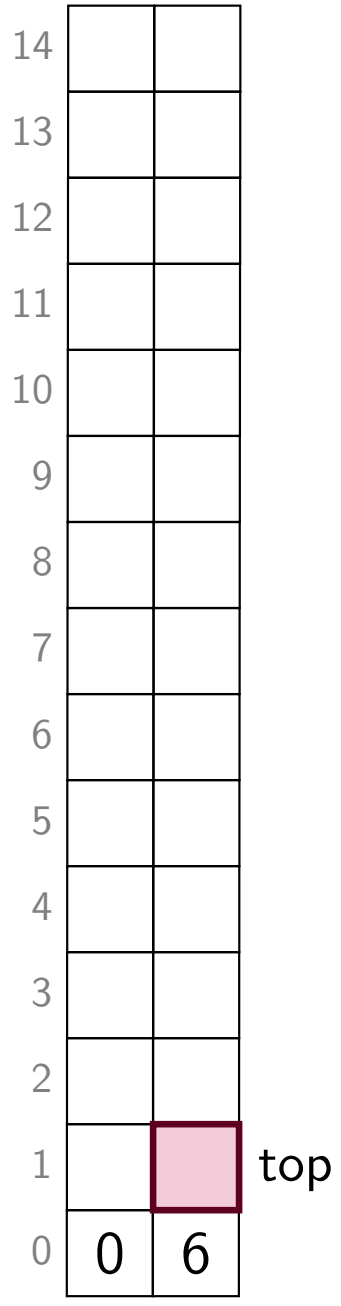
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 8 | 7 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a b

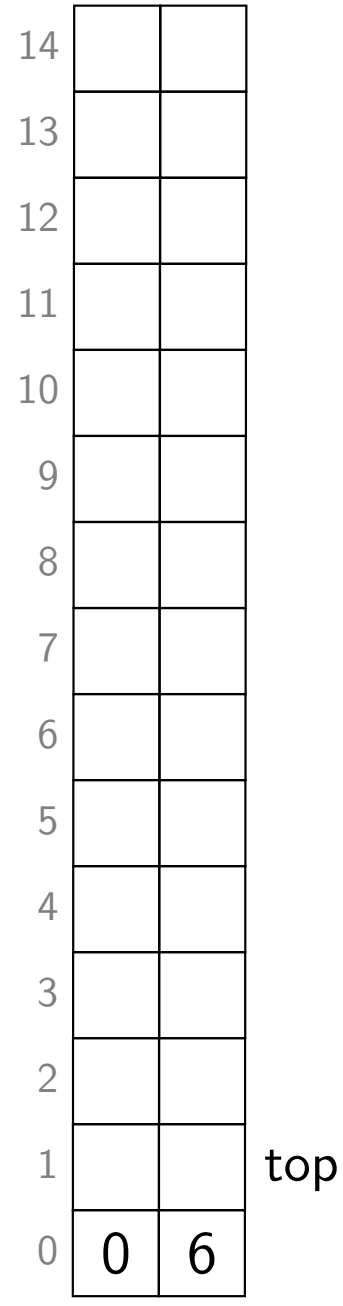
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 8 | 7 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



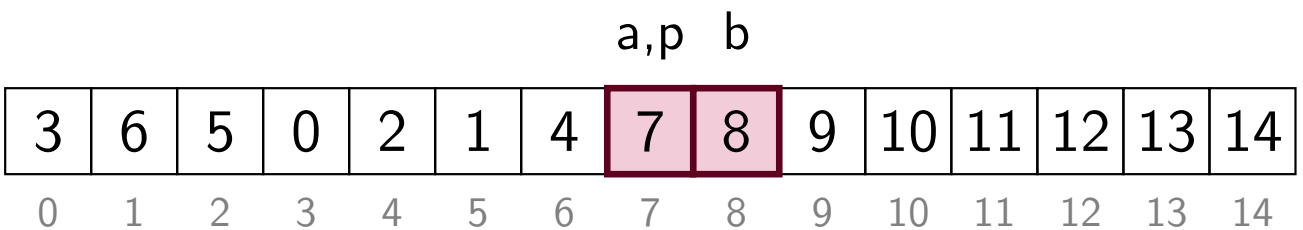
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

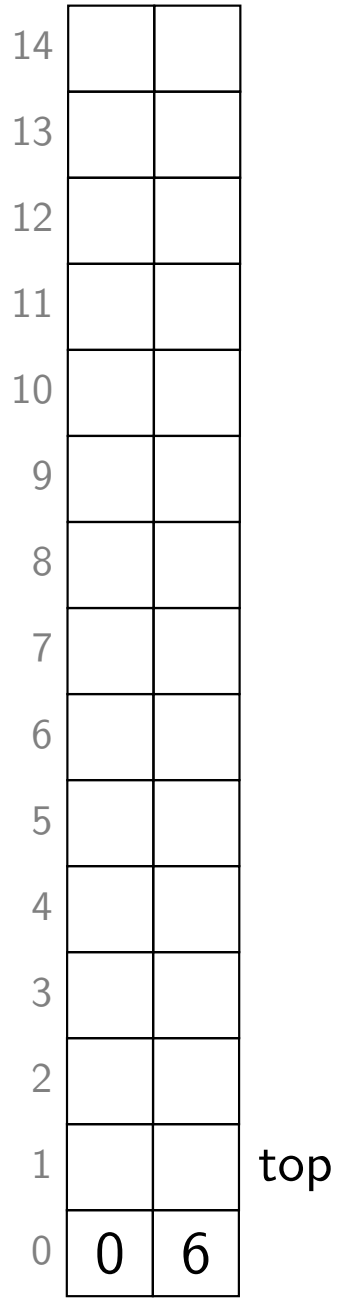
p a b

| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 8 | 7 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

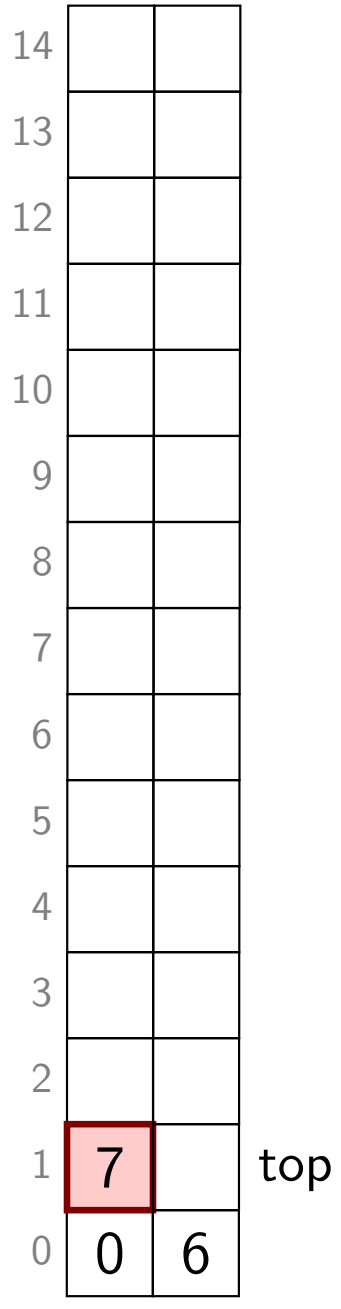


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

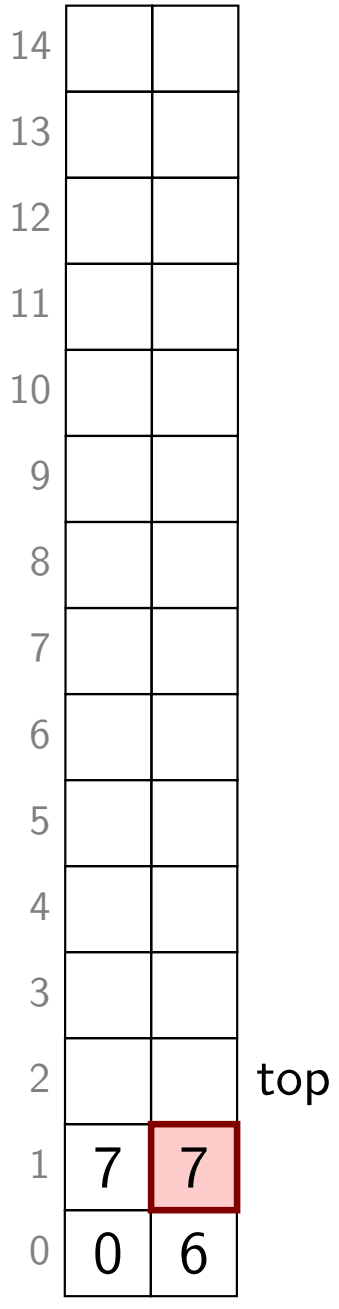
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

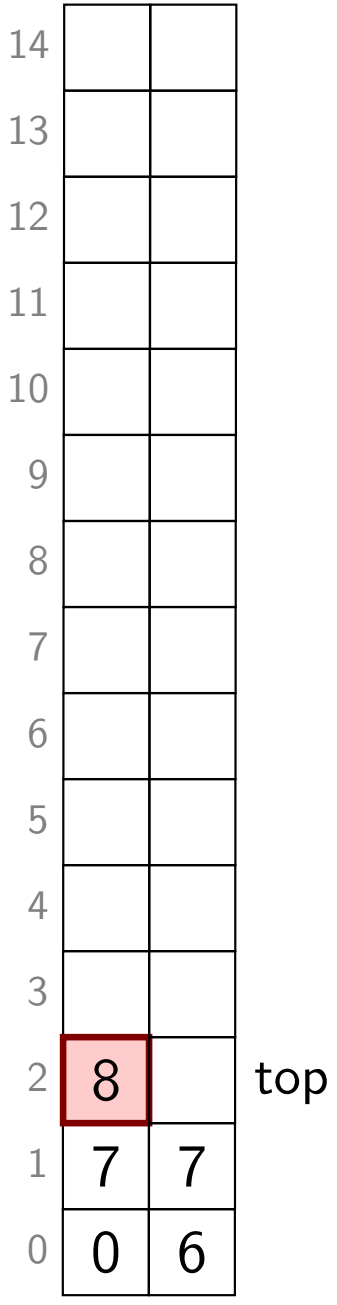
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

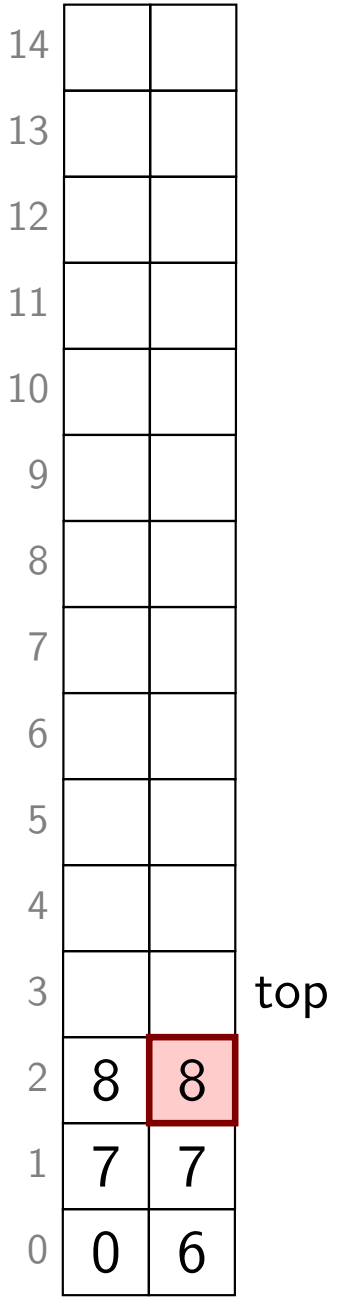
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



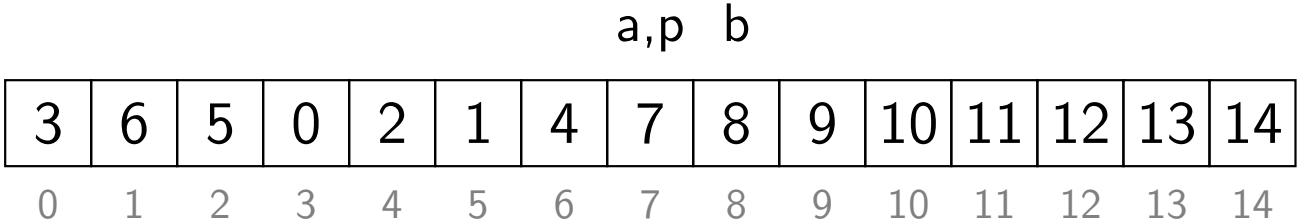
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

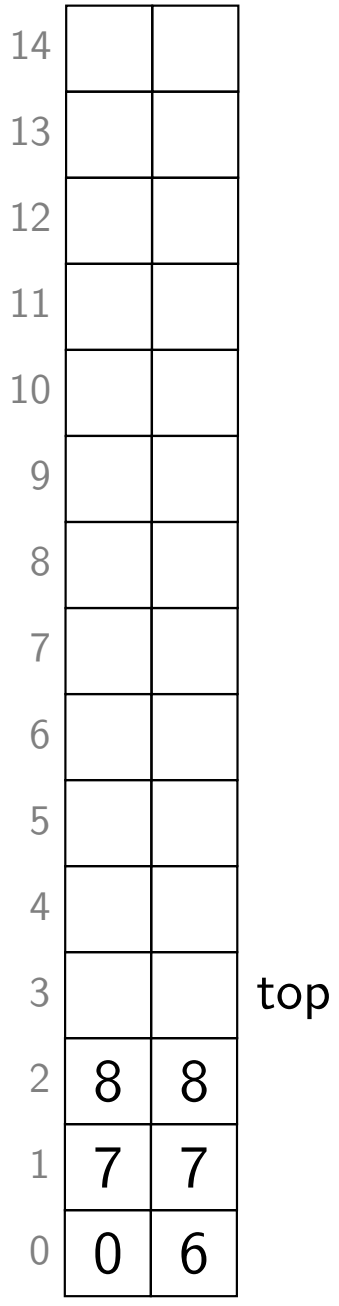
a,p b

| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

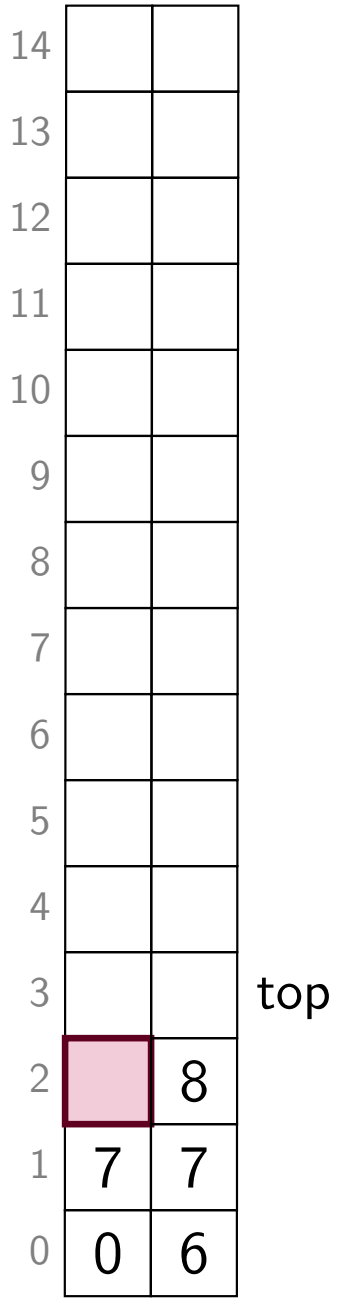


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

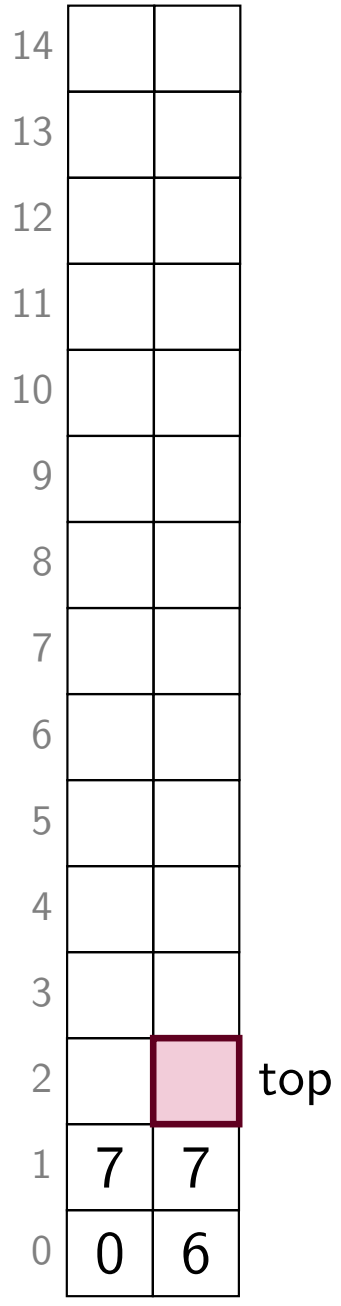
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

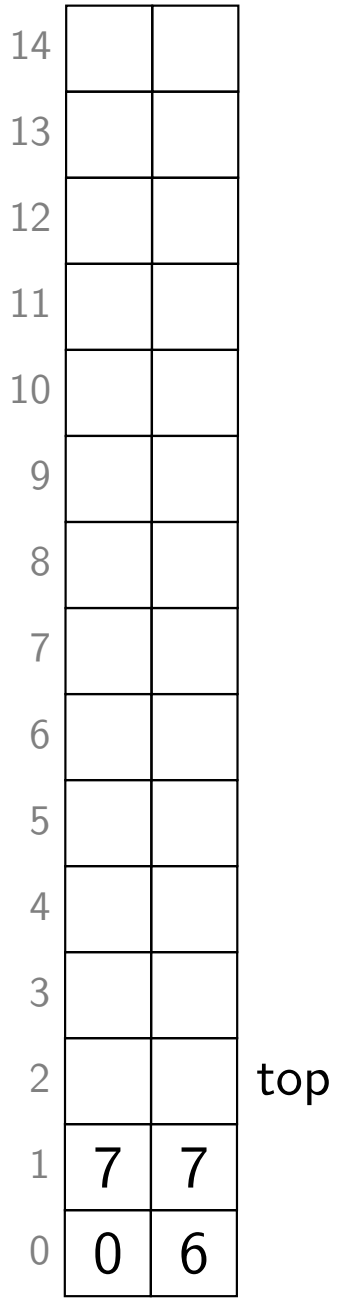
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



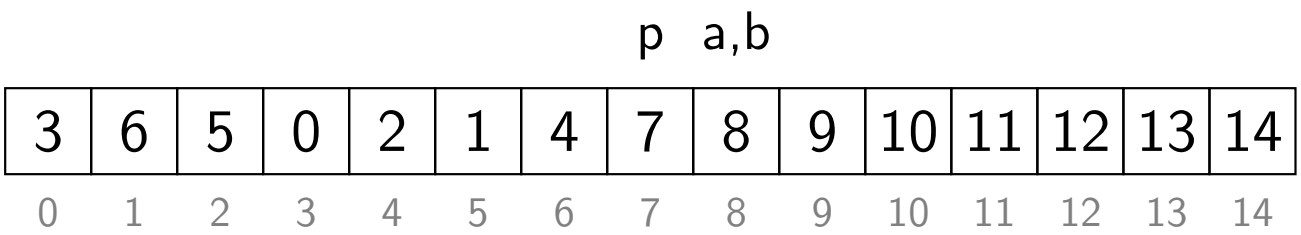
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

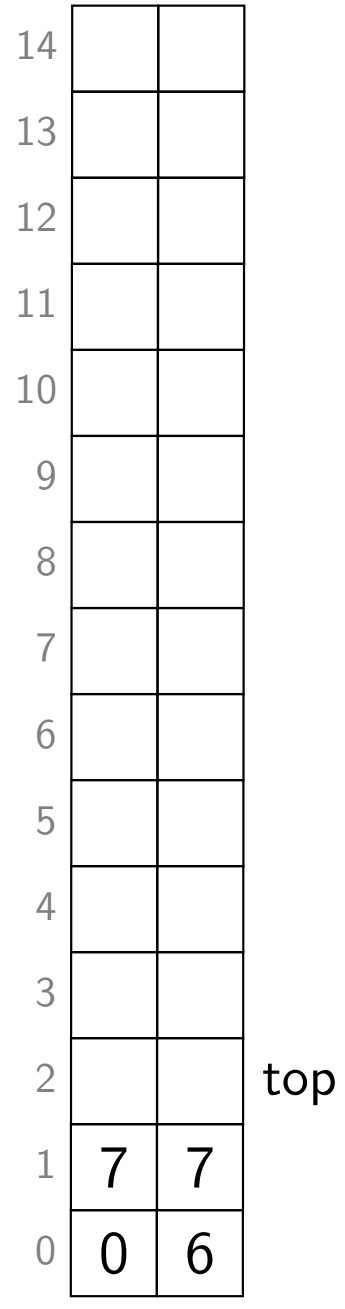
p a,b

| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

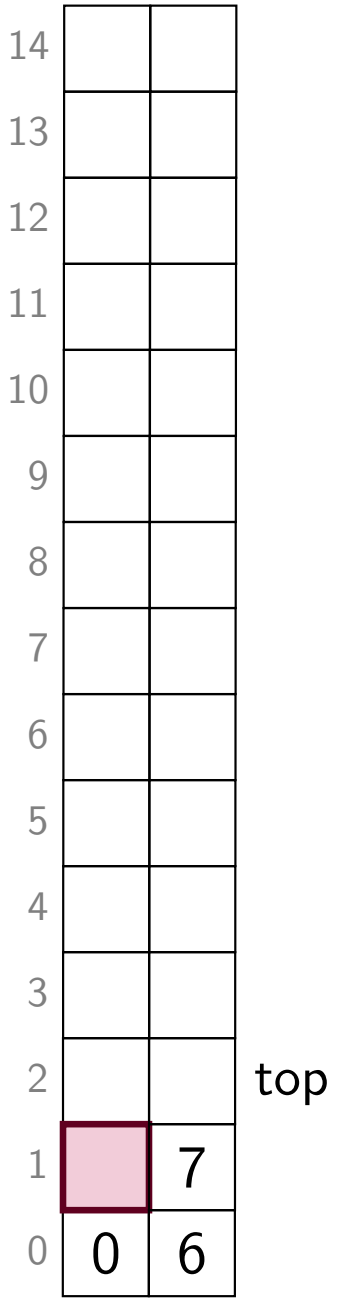


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

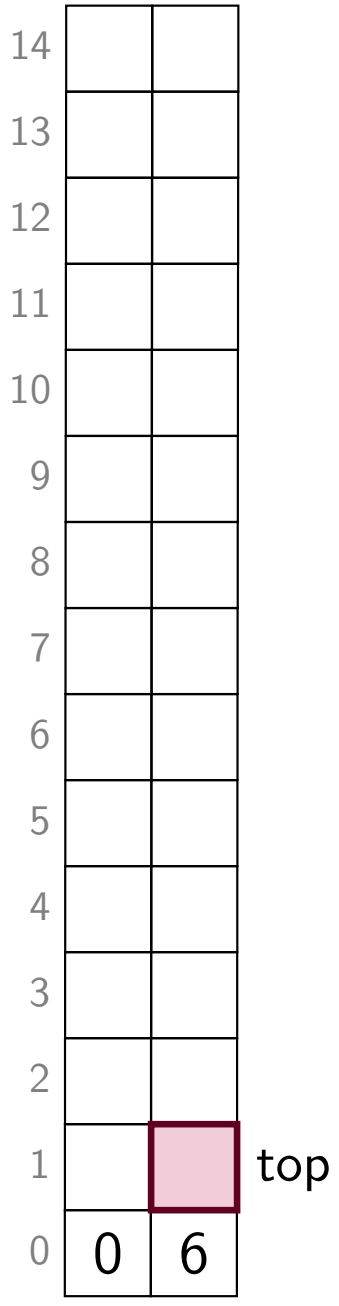
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

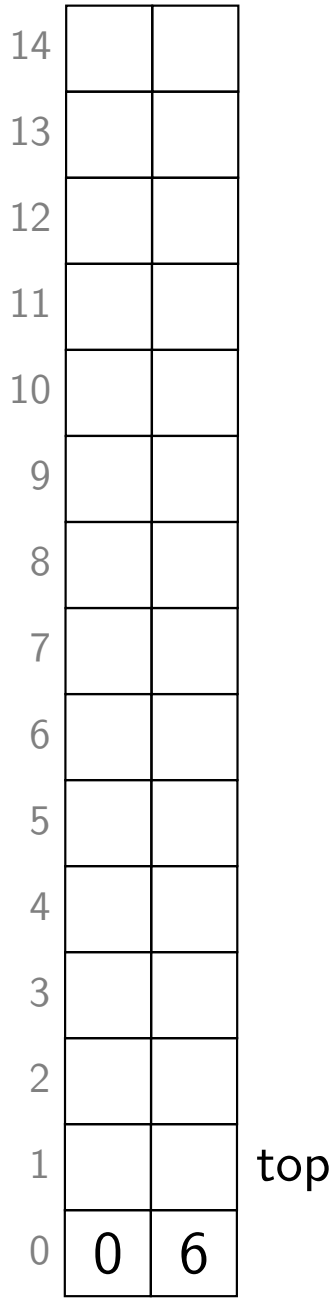
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

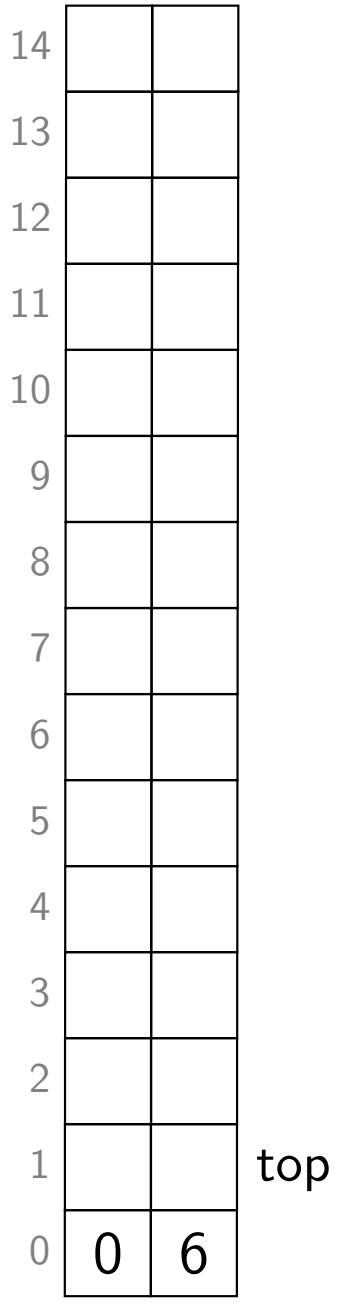
| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



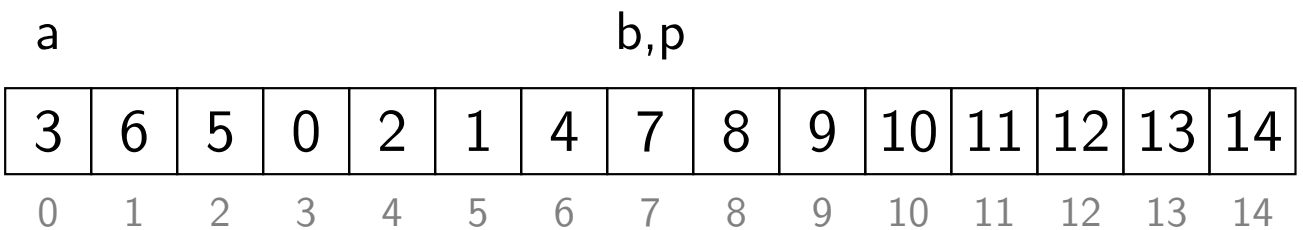
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

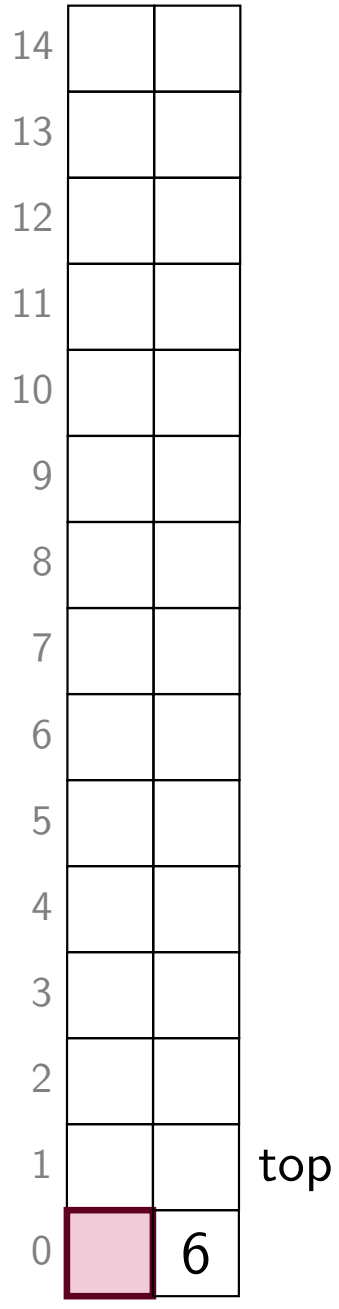
a,b,p

| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

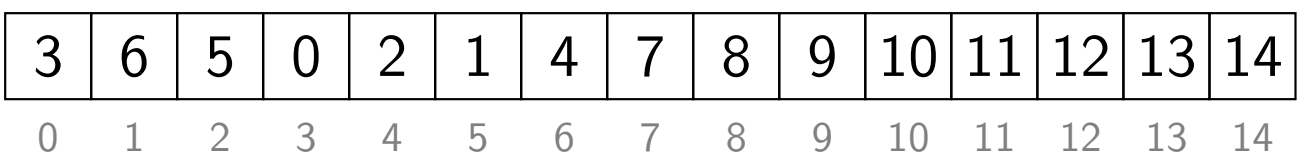


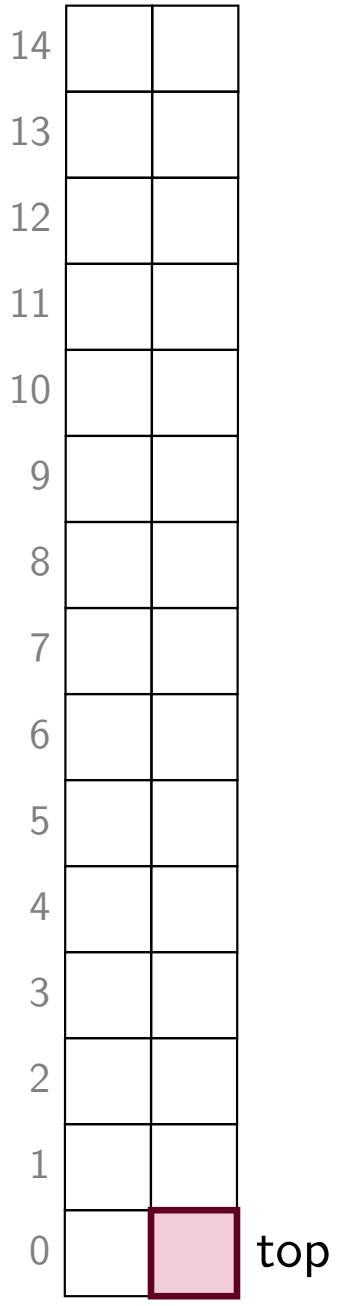
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

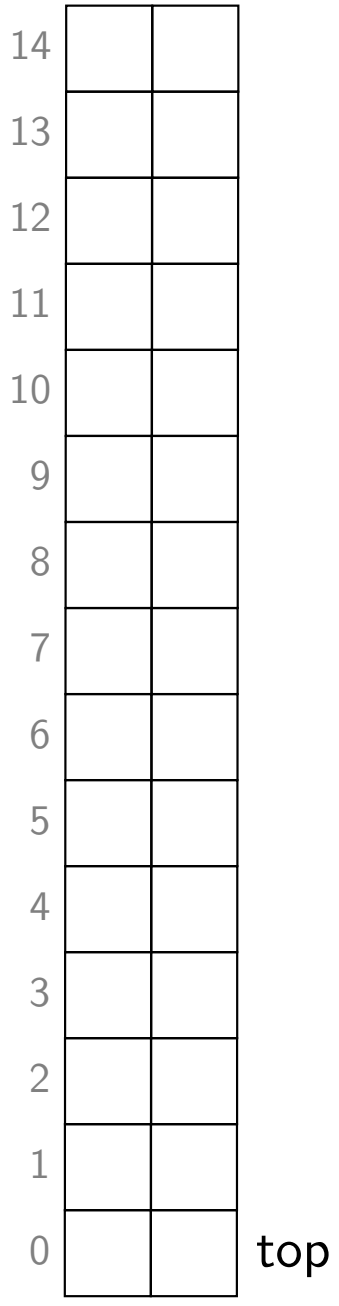


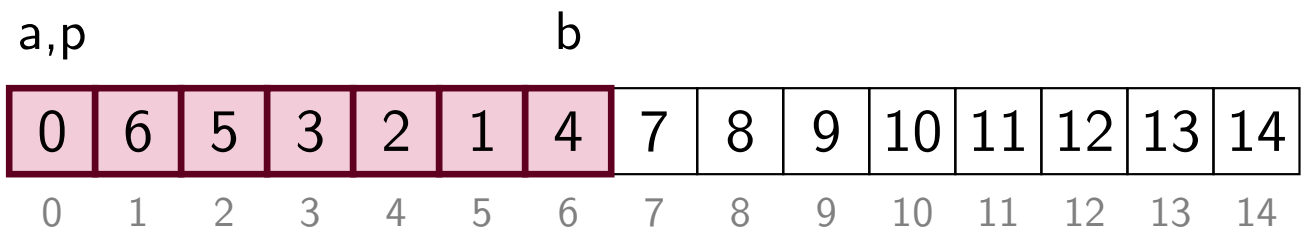
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

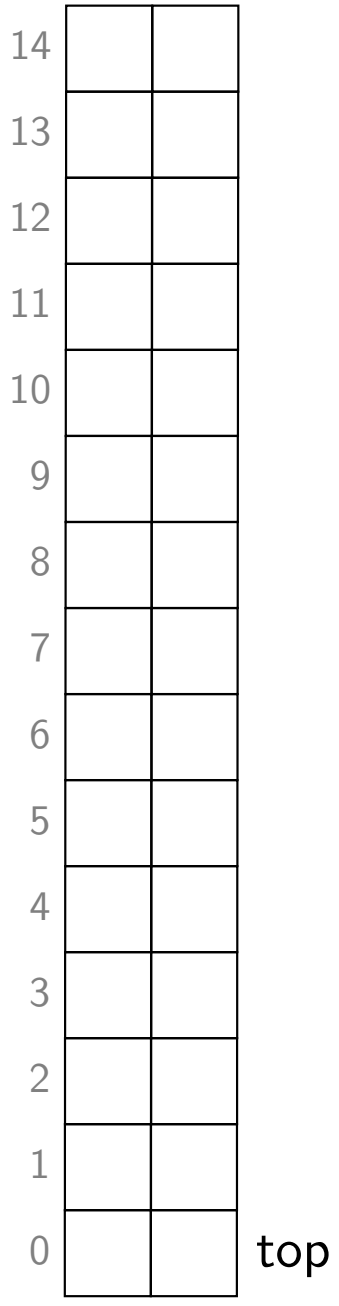
a b p

| 3 | 6 | 5 | 0 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

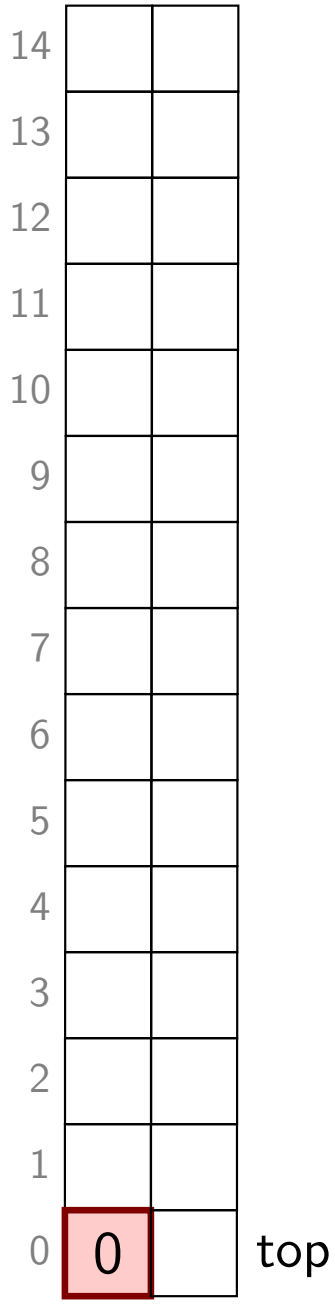


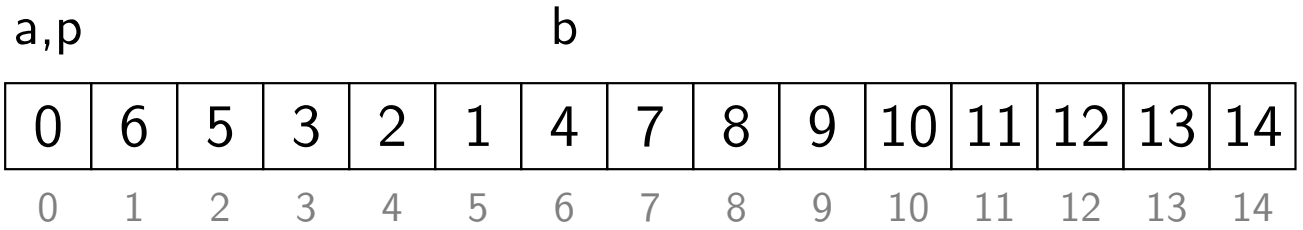
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

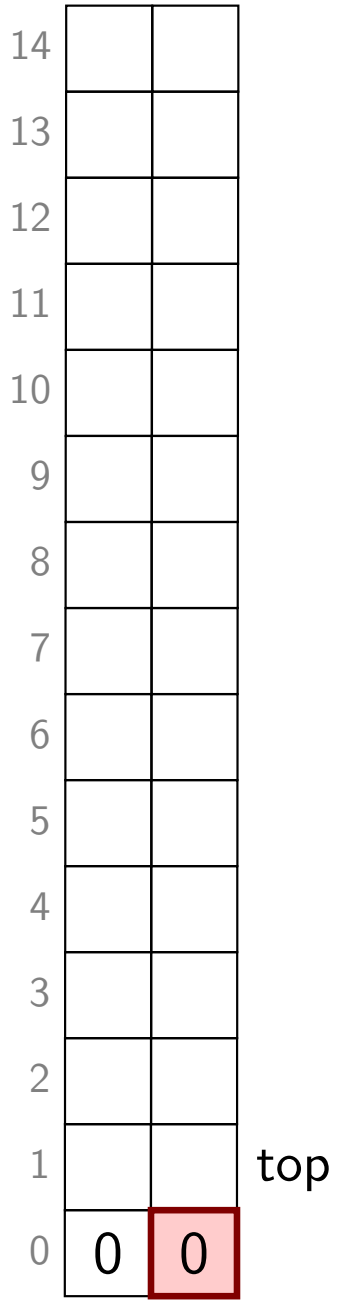
a,p b

| 0 | 6 | 5 | 3 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

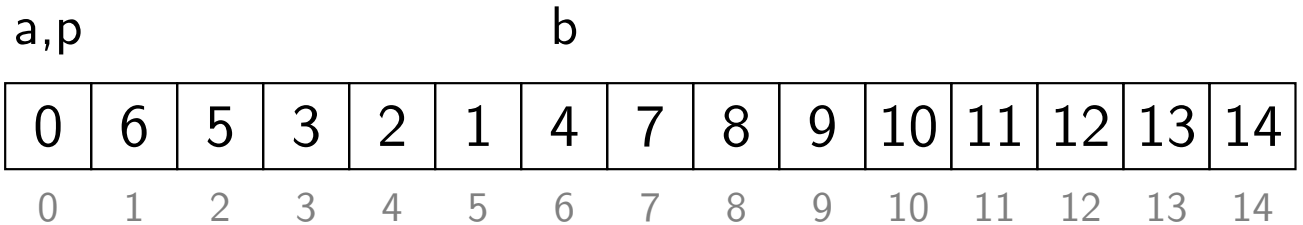


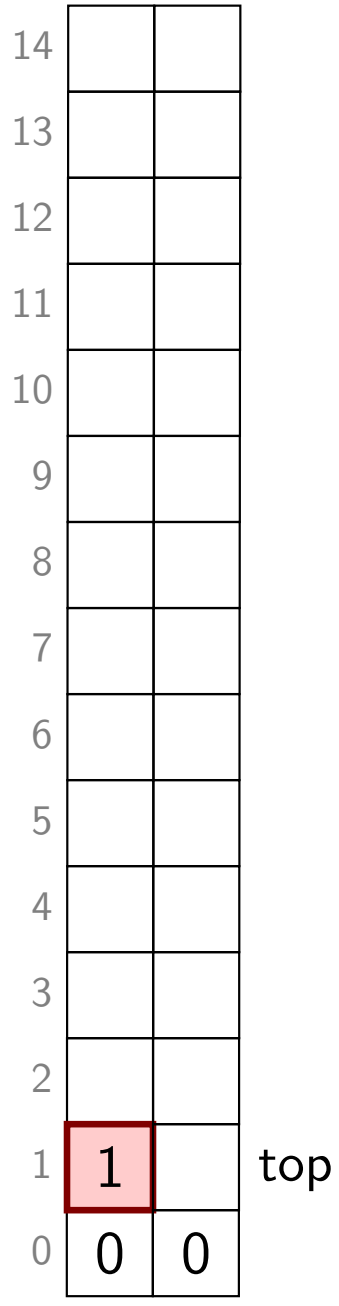
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



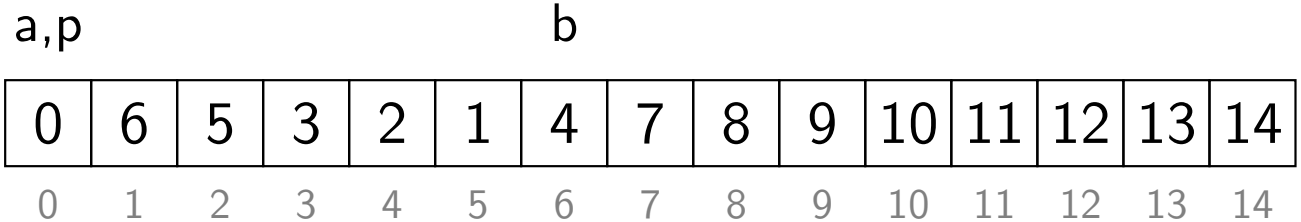



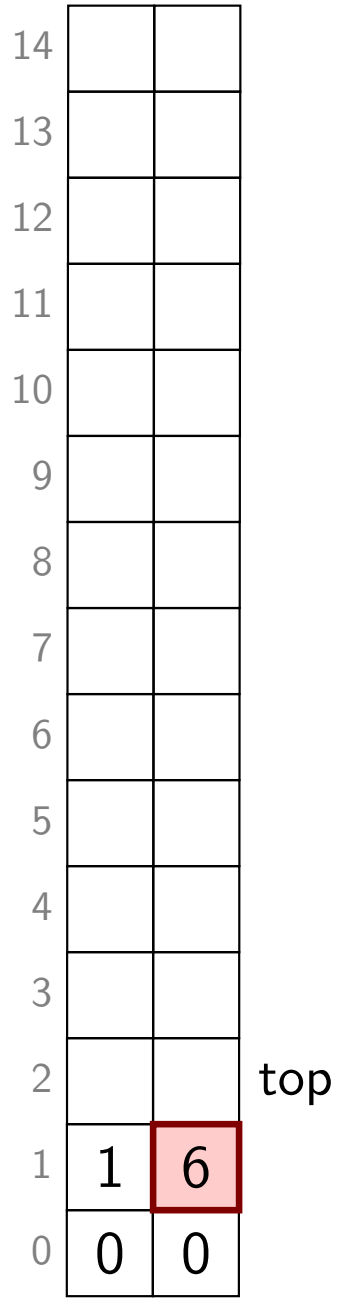
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



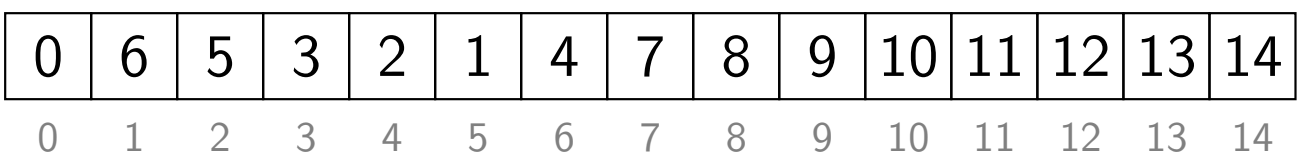



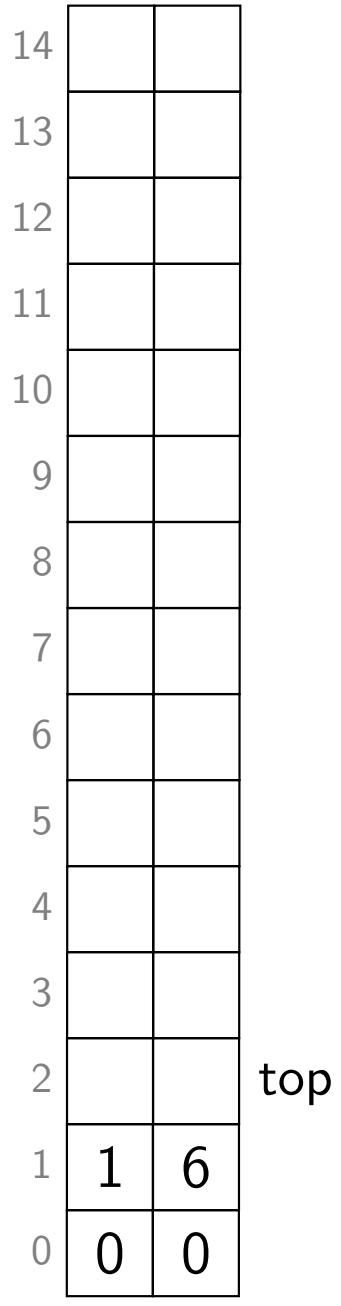
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



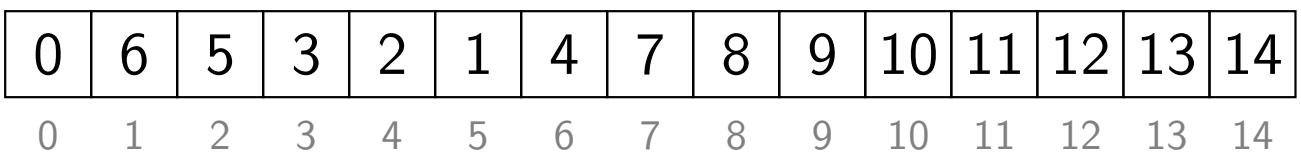



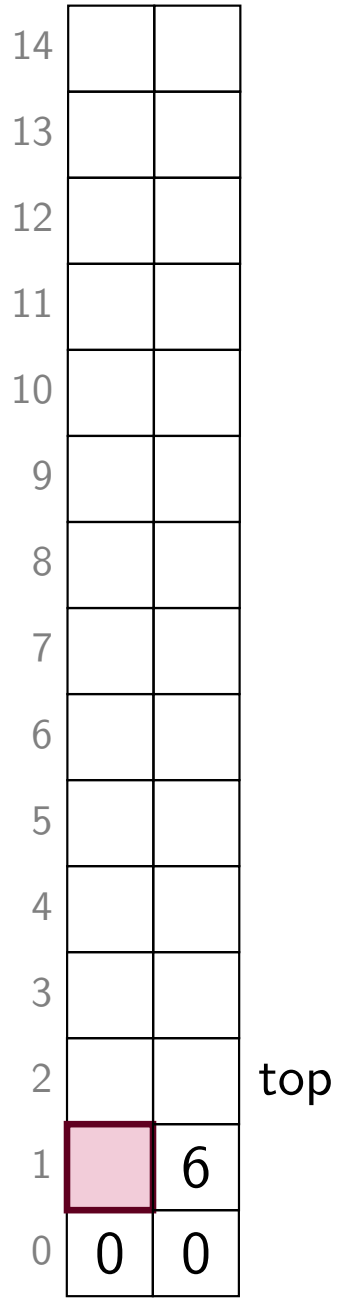
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

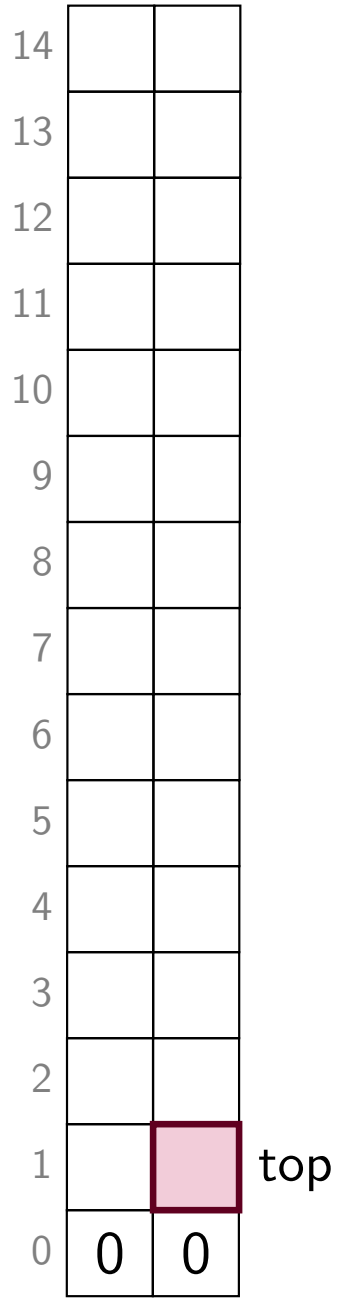


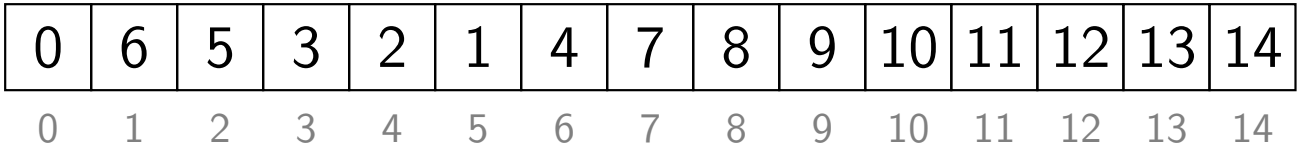
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

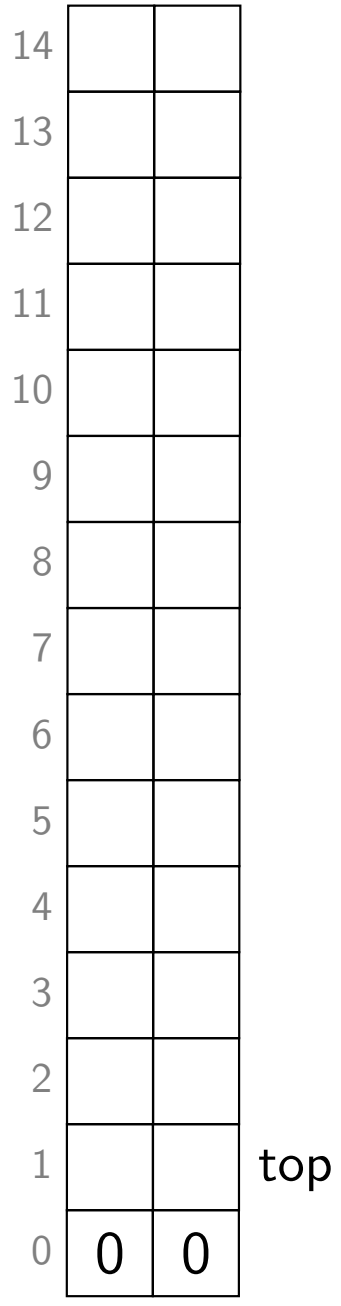
p a b

| 0 | 6 | 5 | 3 | 2 | 1 | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

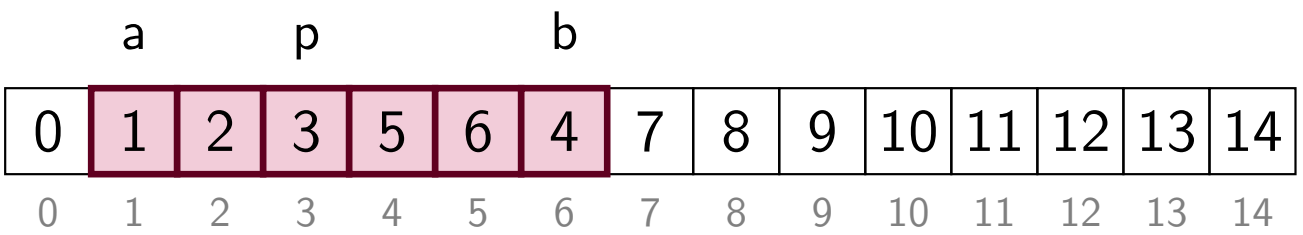


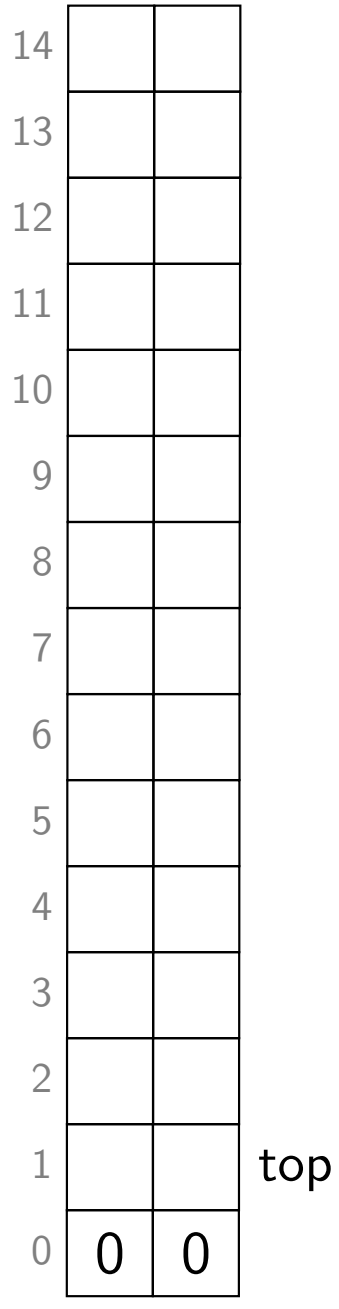
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



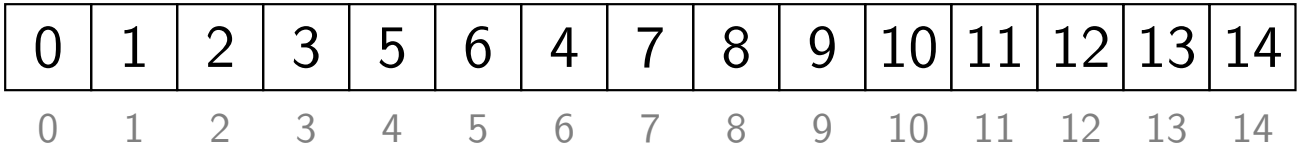



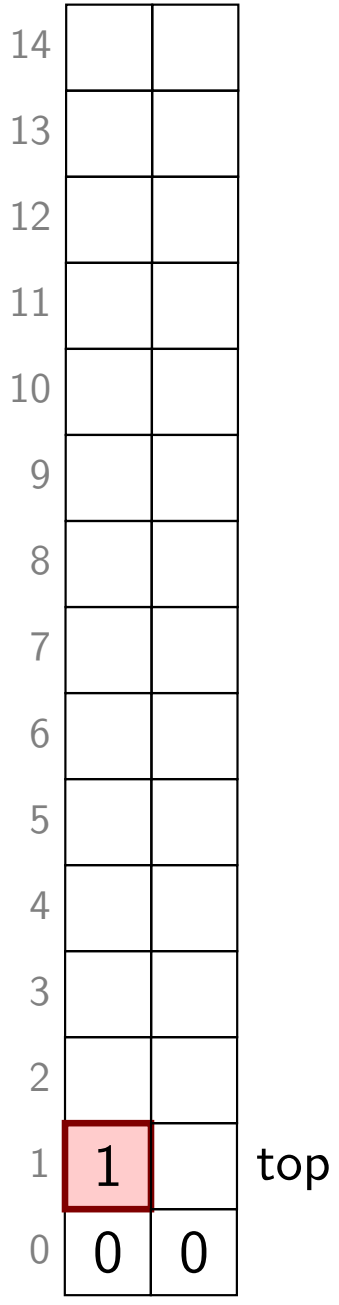
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



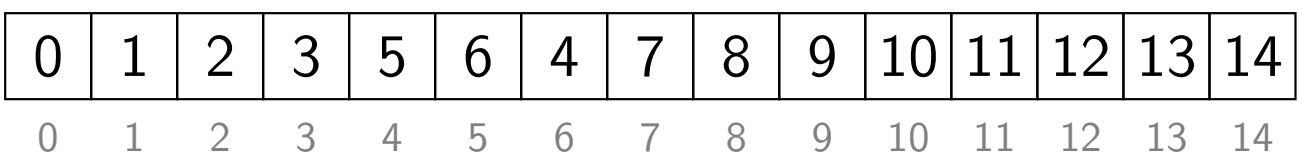



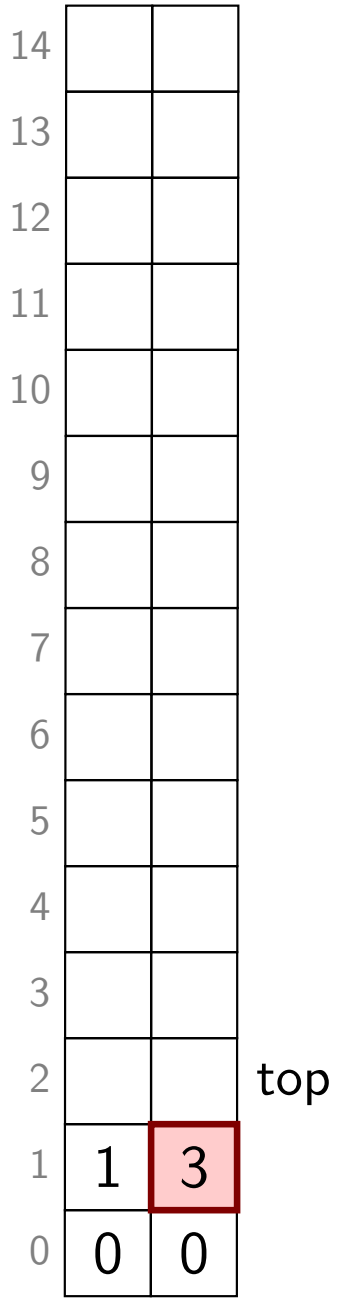
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



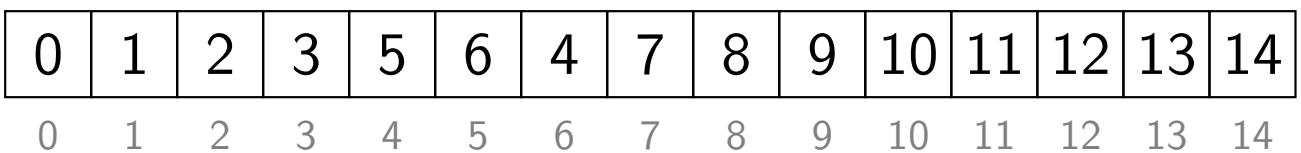



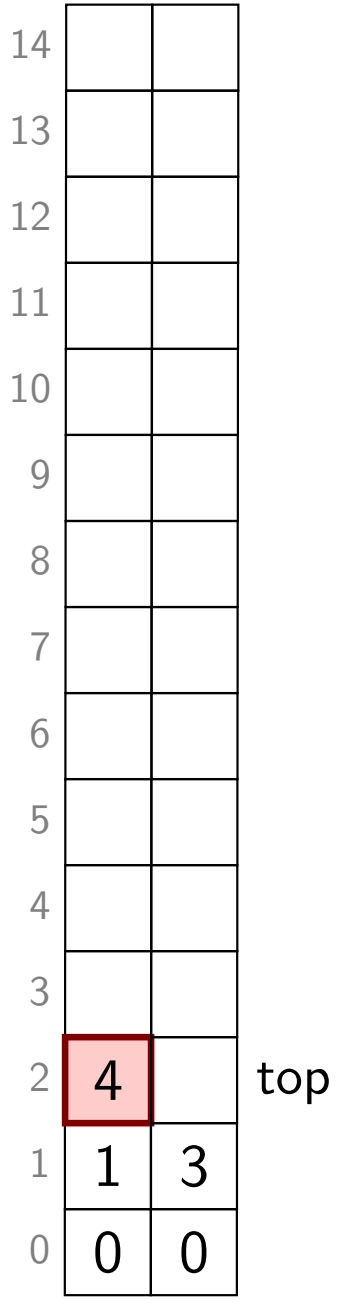
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



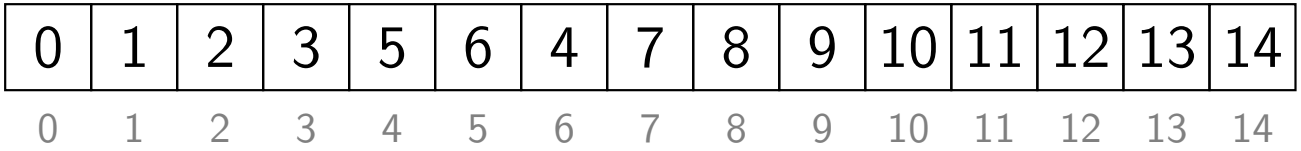



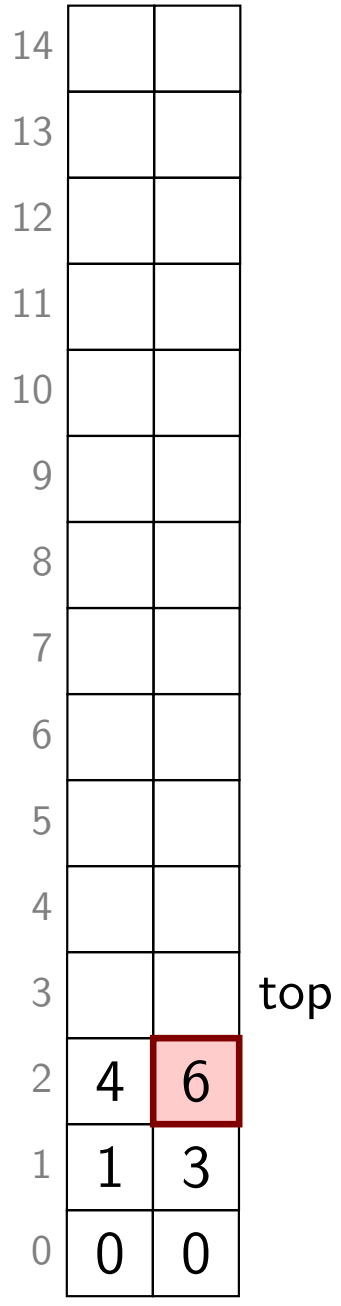
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



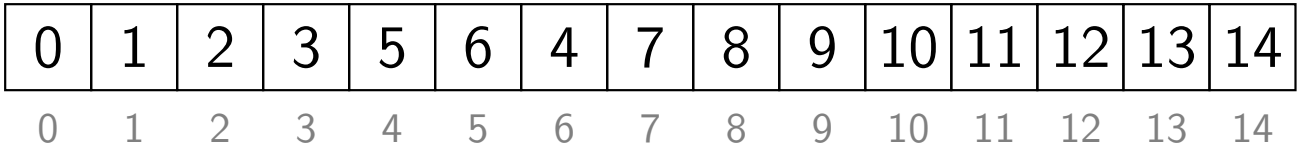



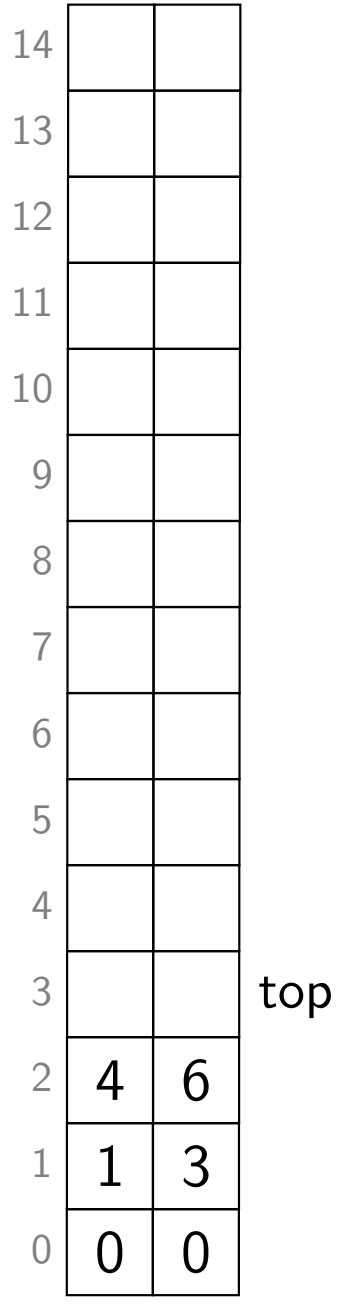
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



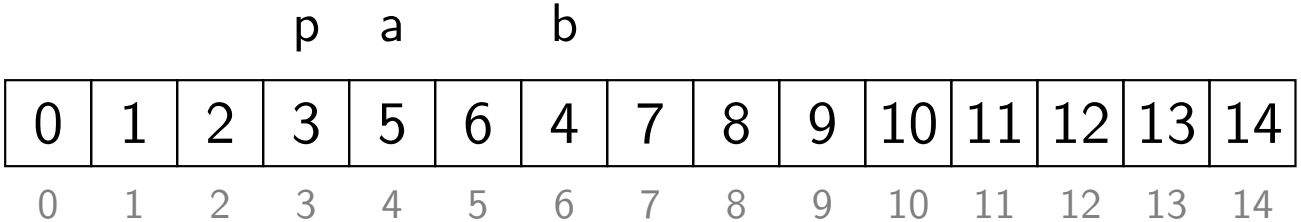



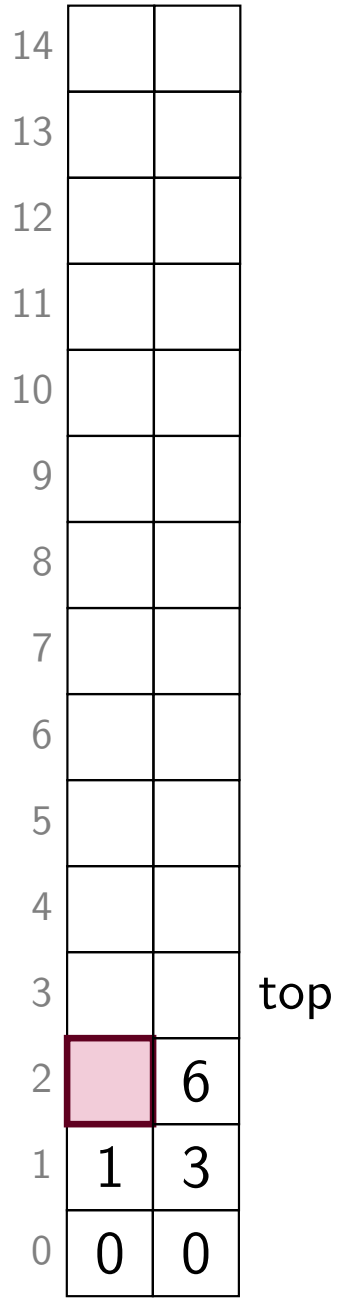
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



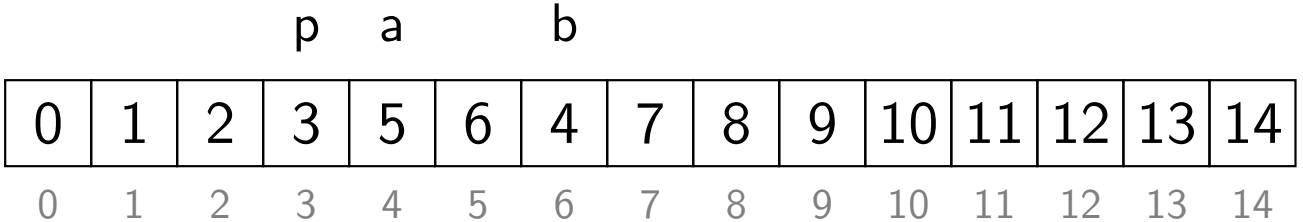



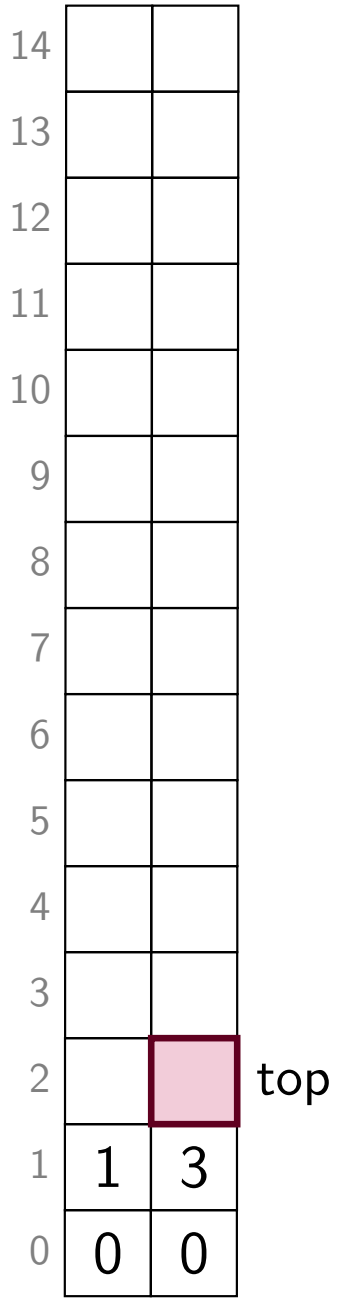
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



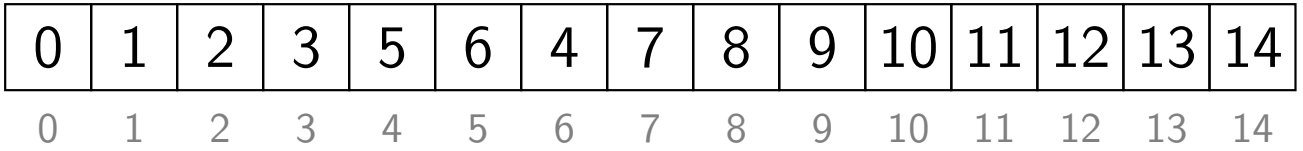



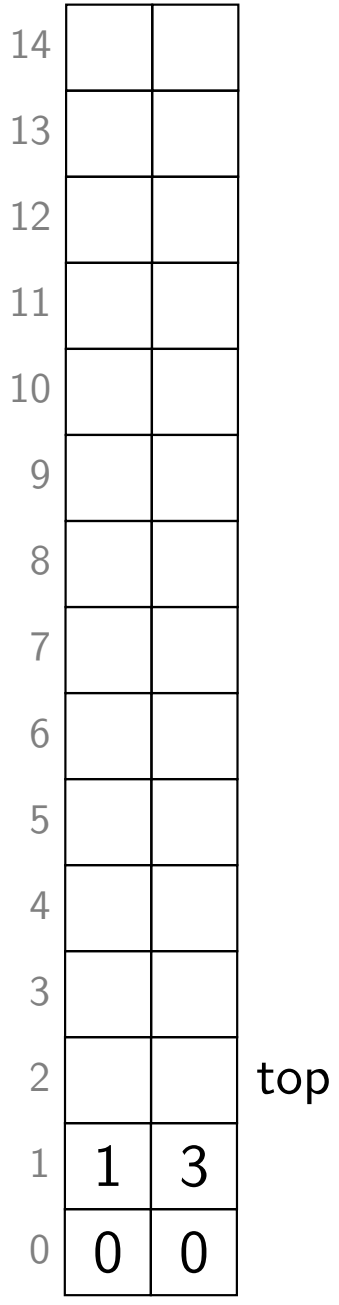
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



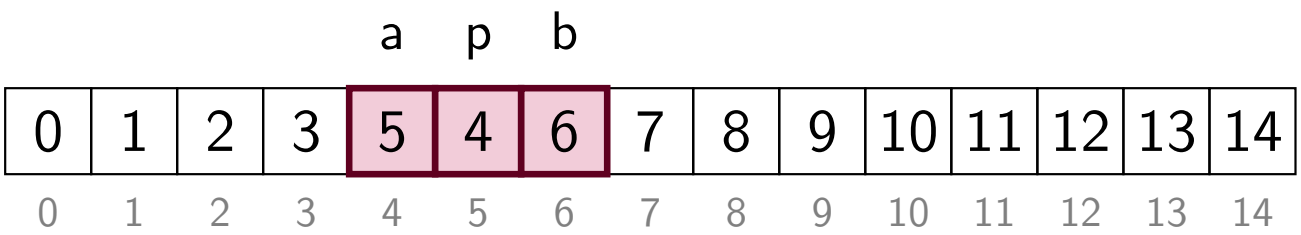



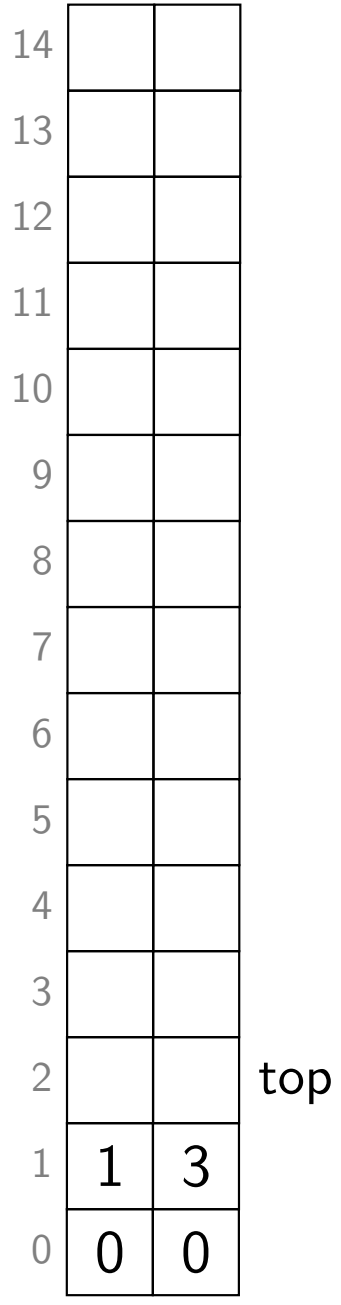
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



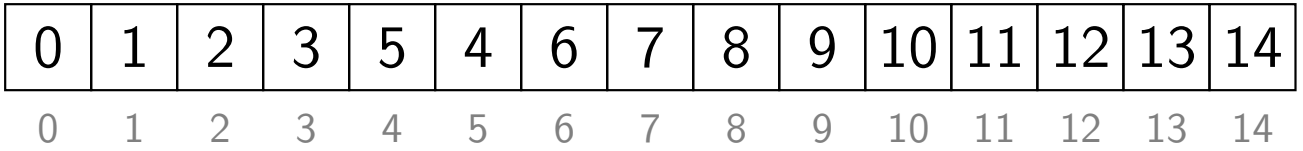



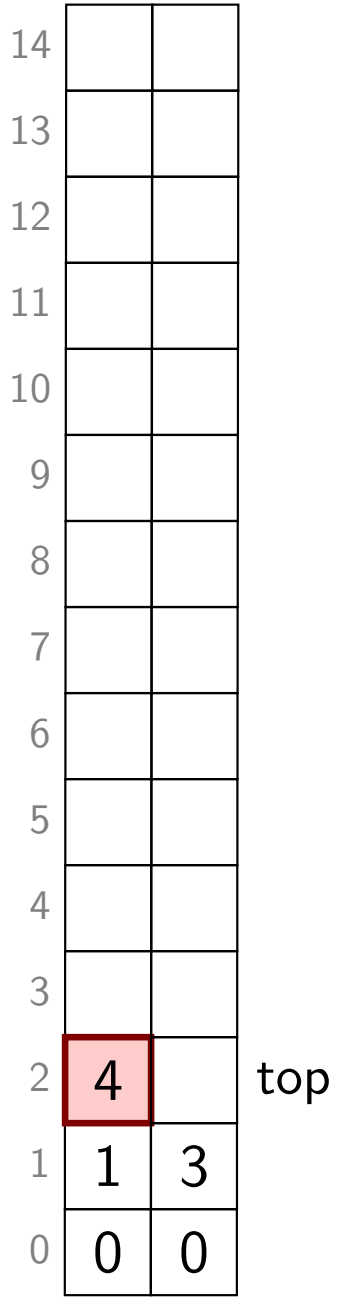
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



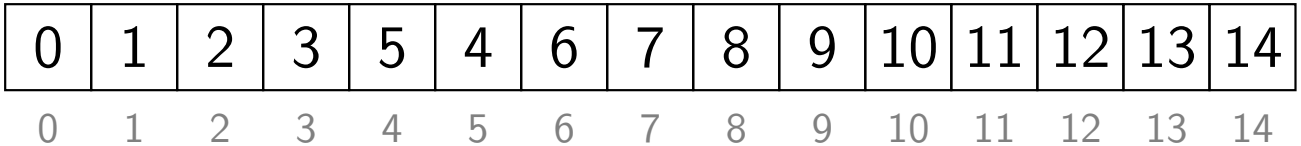



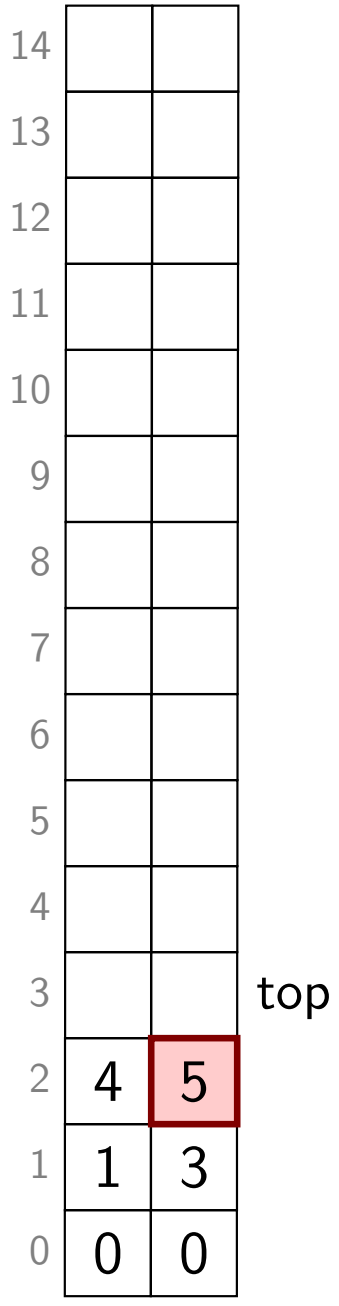
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



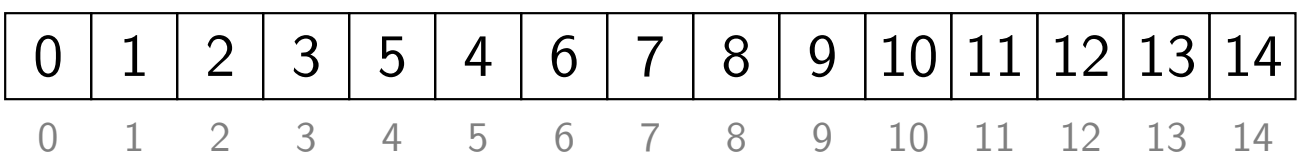



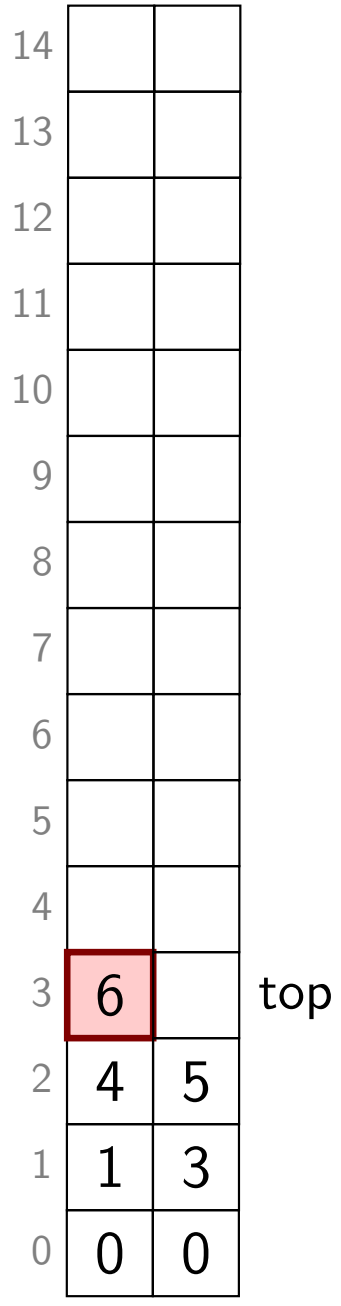
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



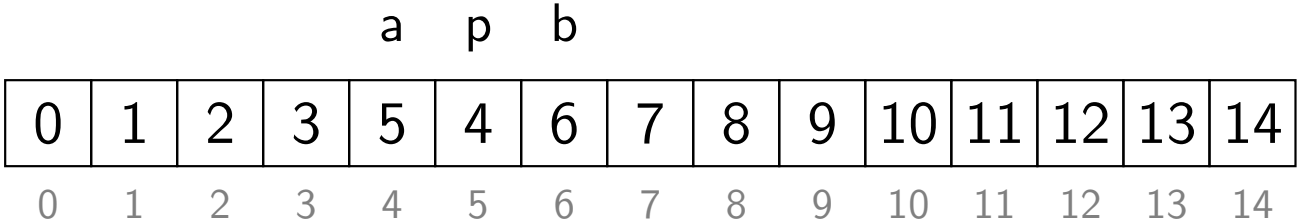



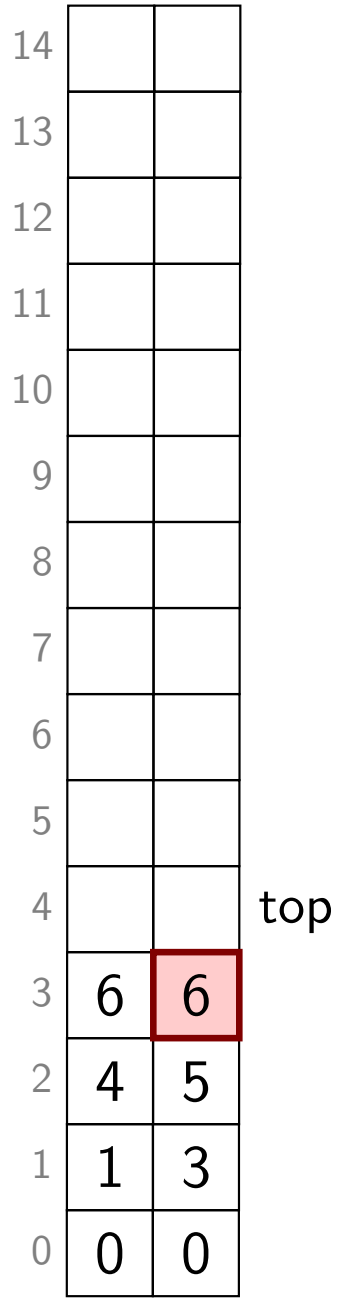
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



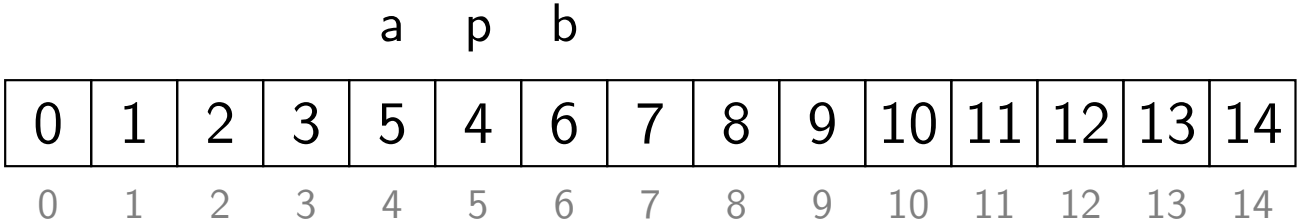



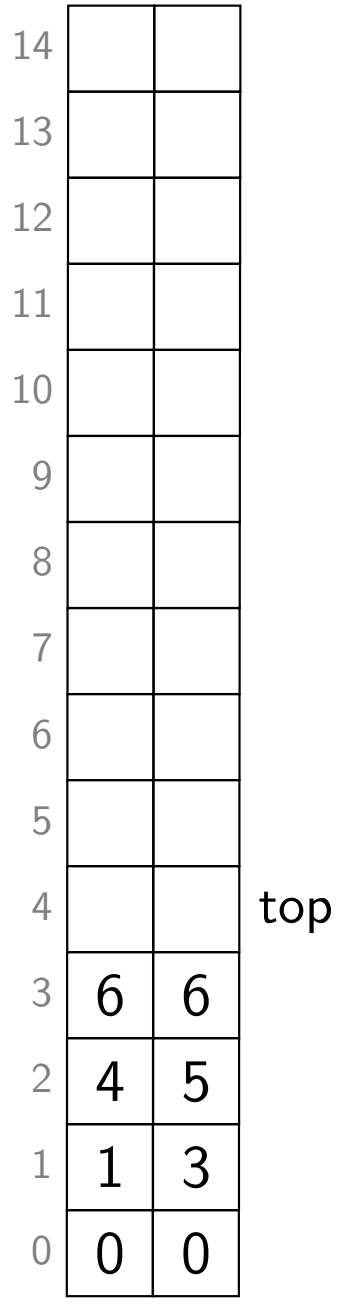
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



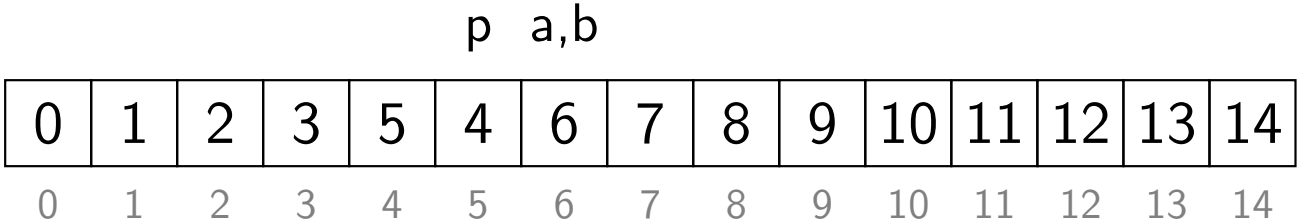



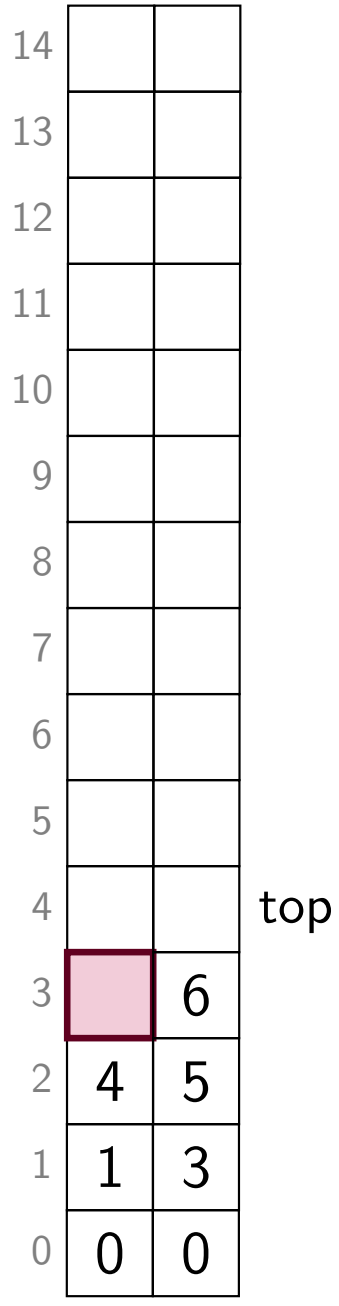
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



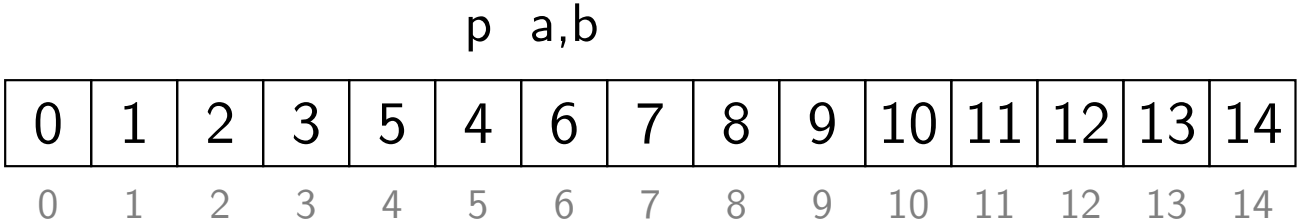



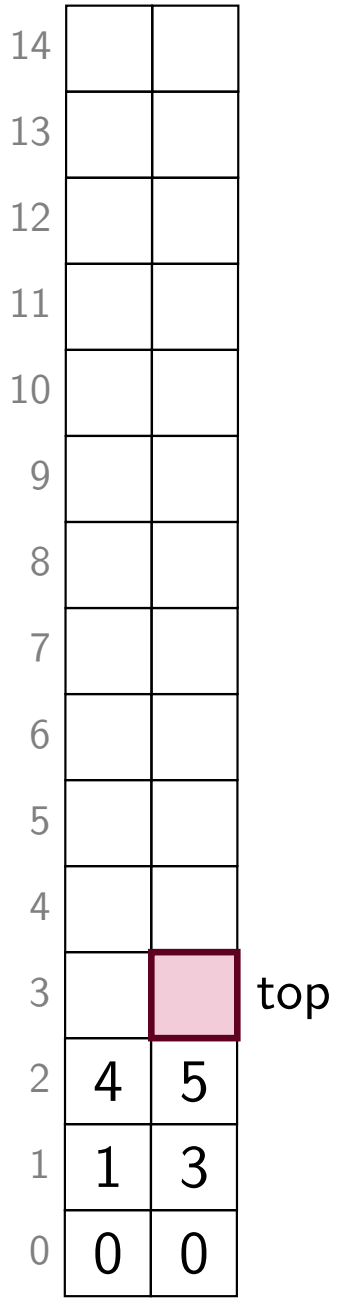
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

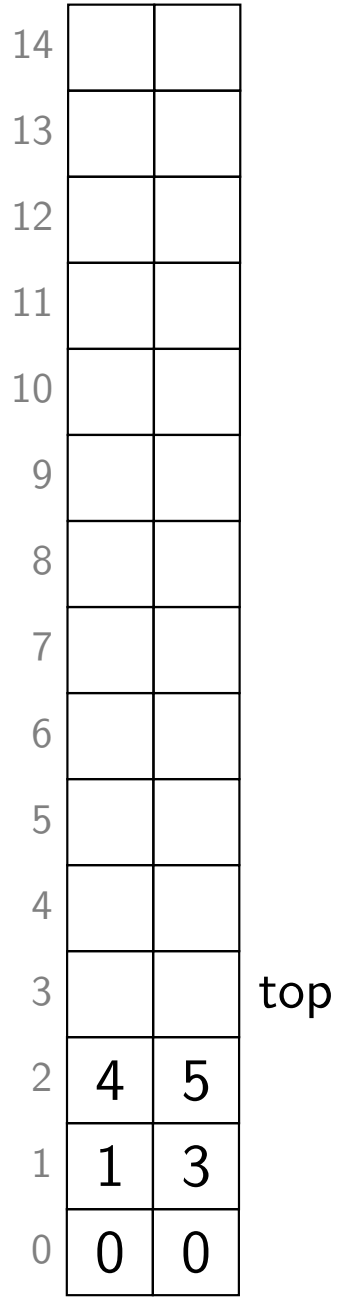
| 0 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| | | | |
|---|---|---|---|
| 14 | | | |
| 13 | | | |
| 12 | | | |
| 11 | | | |
| 10 | | | |
| 9 | | | |
| 8 | | | |
| 7 | | | |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | | | top |
| 2 | 4 | 5 | |
| 1 | 1 | 3 | |
| 0 | 0 | 0 | |

```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

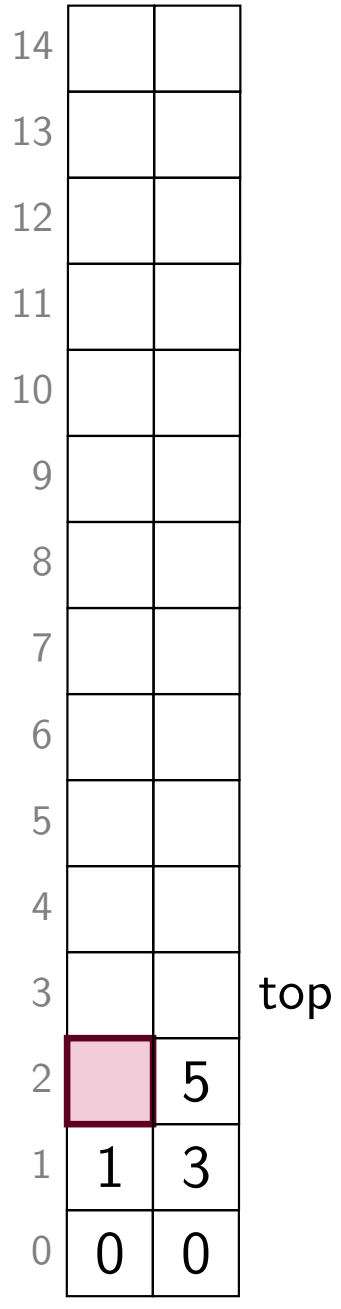
| 0 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



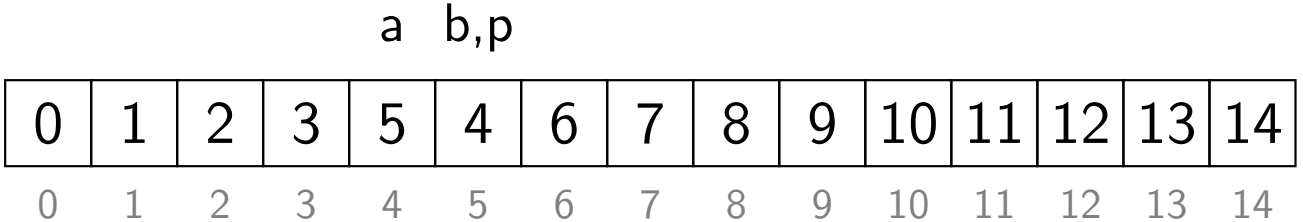
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

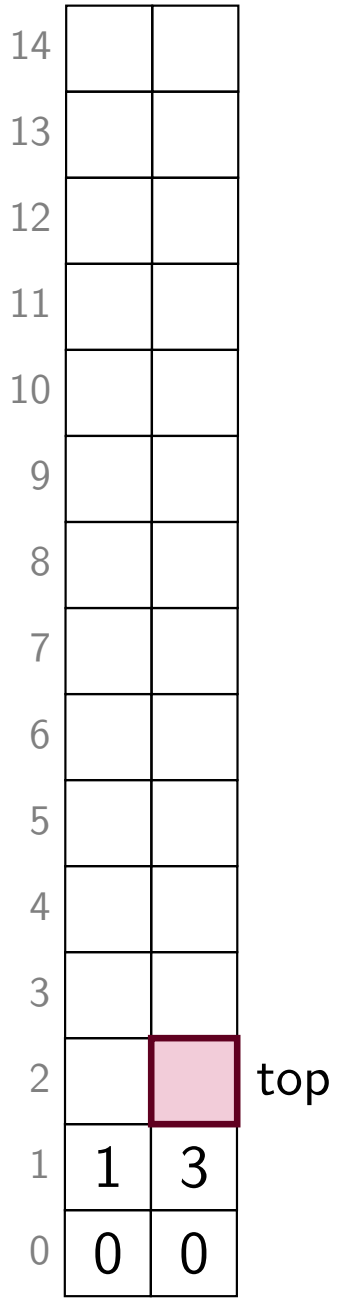
a p b

| 0 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



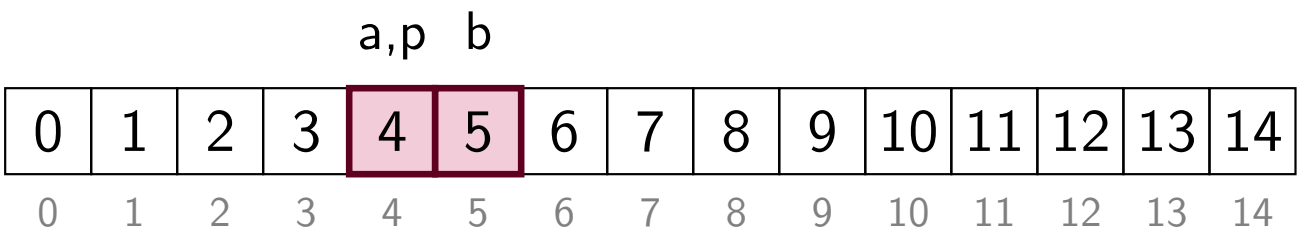
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

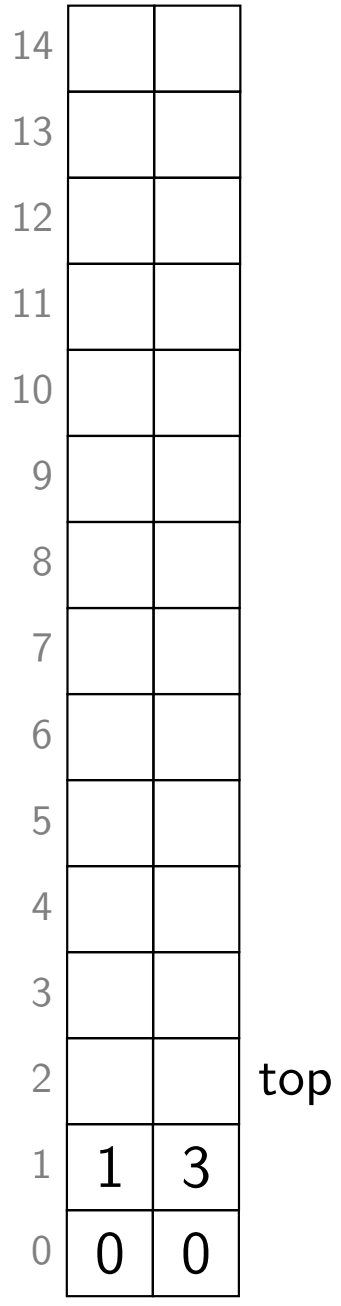
a b,p

| 0 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

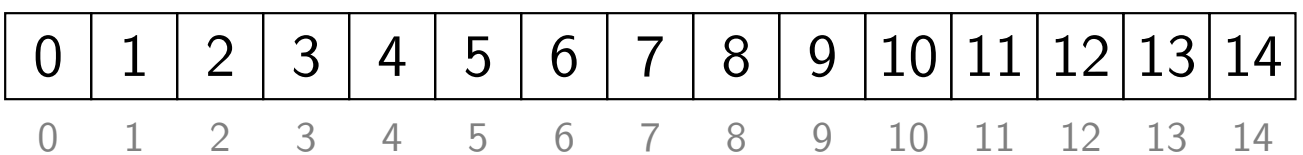
| | | | |
|---|---|---|---|
| 14 | | | |
| 13 | | | |
| 12 | | | |
| 11 | | | |
| 10 | | | |
| 9 | | | |
| 8 | | | |
| 7 | | | |
| 6 | | | |
| 5 | | | |
| 4 | | | |
| 3 | | | |
| 2 | | | top |
| 1 | 1 | 3 | |
| 0 | 0 | 0 | |

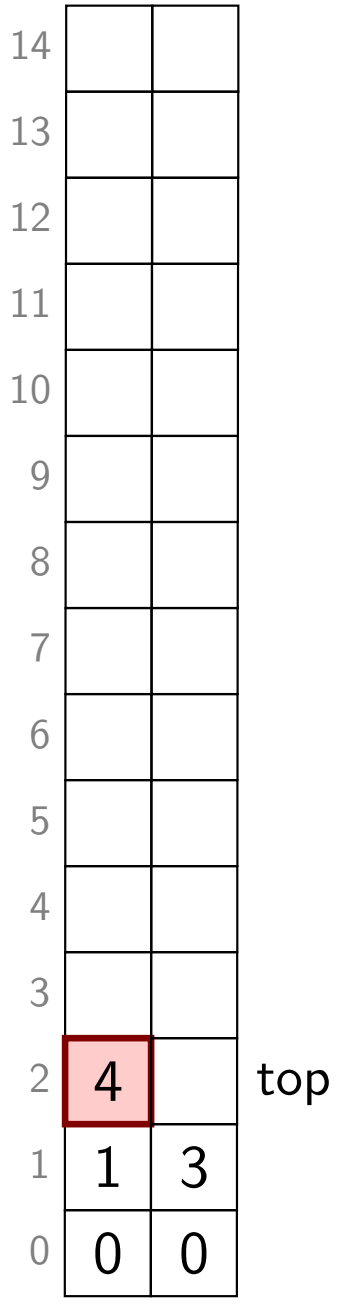
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



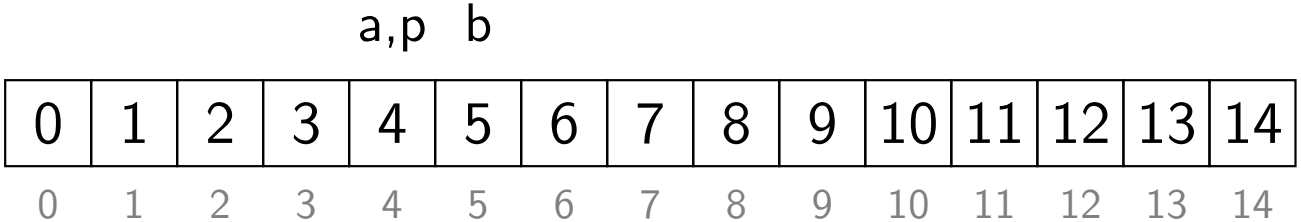



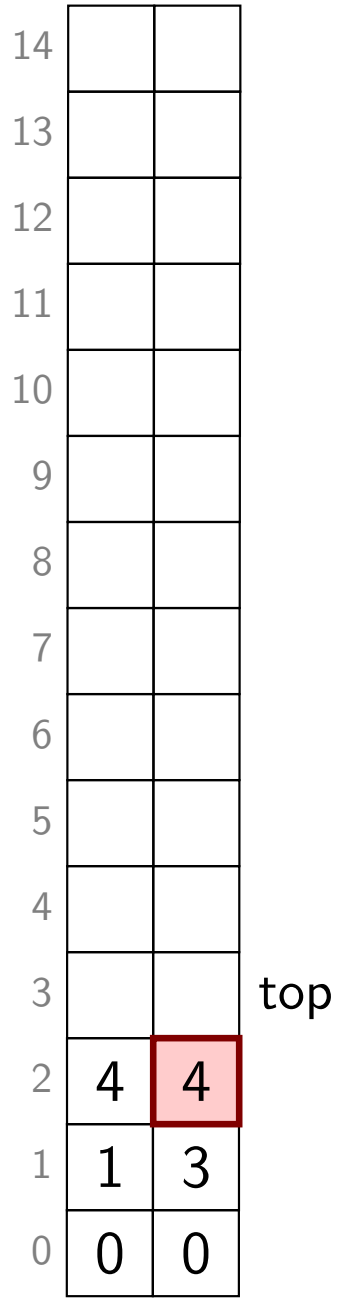
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



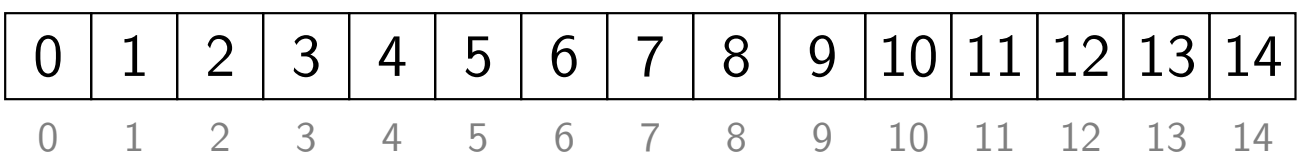



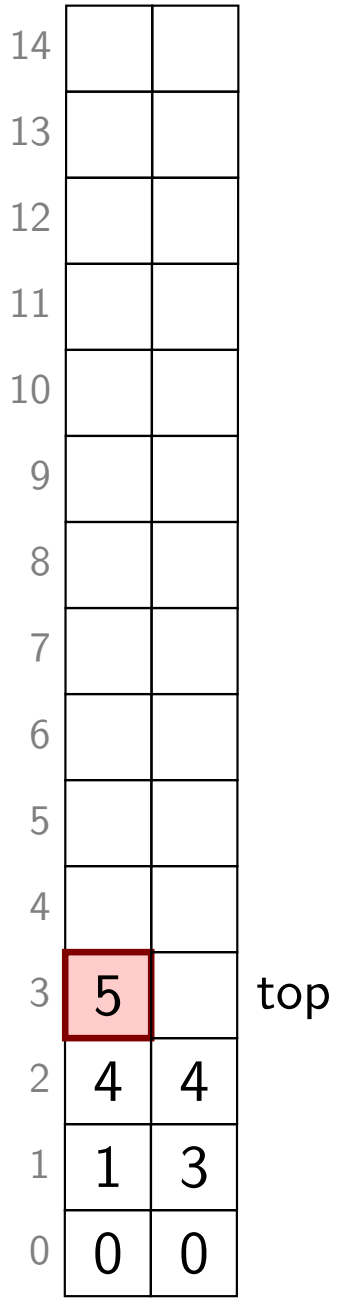
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

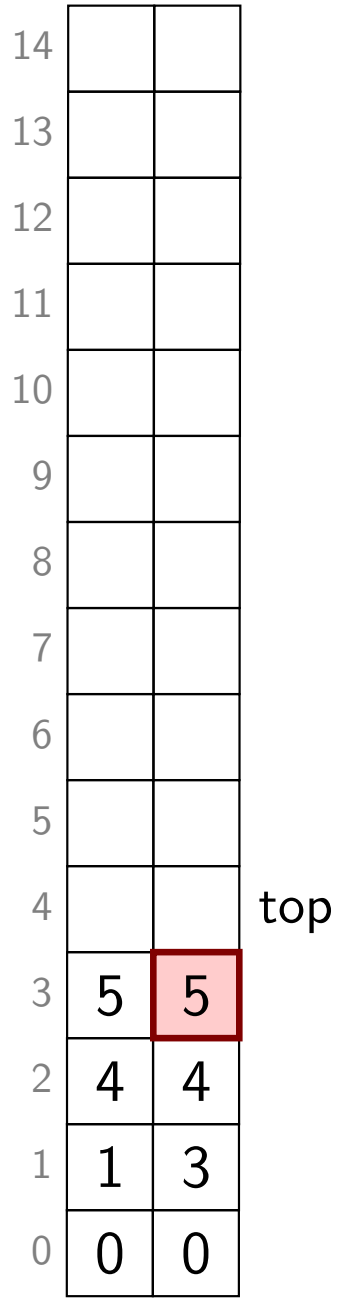


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

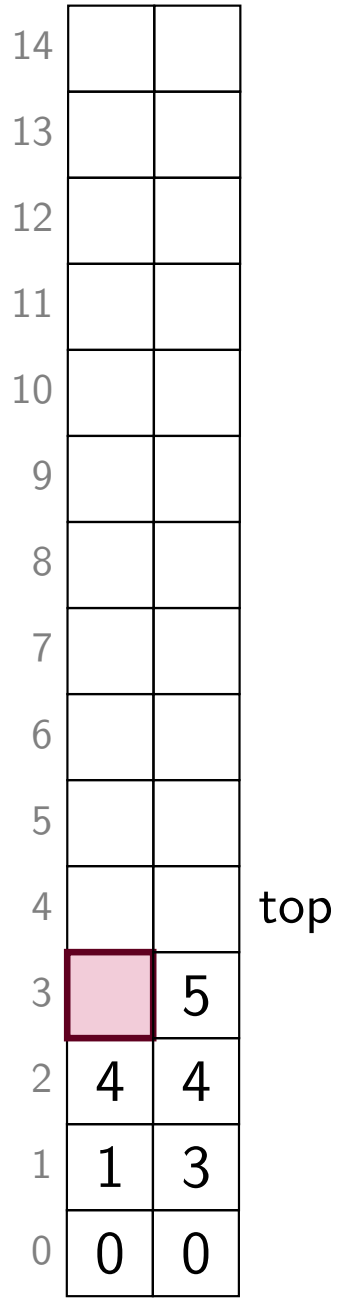
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| | | |
|---|---|---|
| 14 | | |
| 13 | | |
| 12 | | |
| 11 | | |
| 10 | | |
| 9 | | |
| 8 | | |
| 7 | | |
| 6 | | |
| 5 | | |
| 4 | | | top |
| 3 | 5 | 5 |
| 2 | 4 | 4 |
| 1 | 1 | 3 |
| 0 | 0 | 0 |

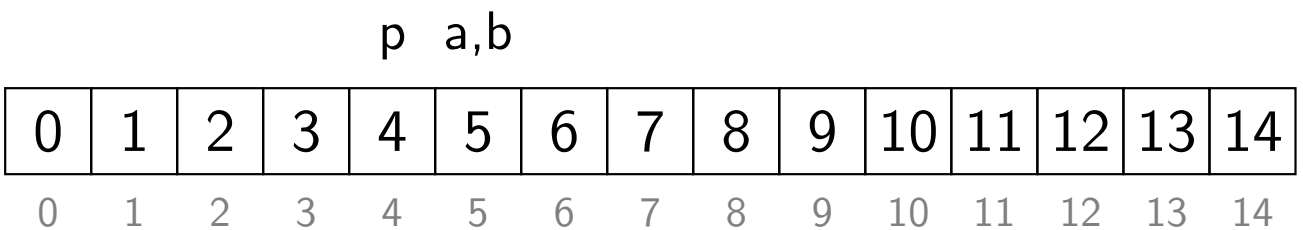
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

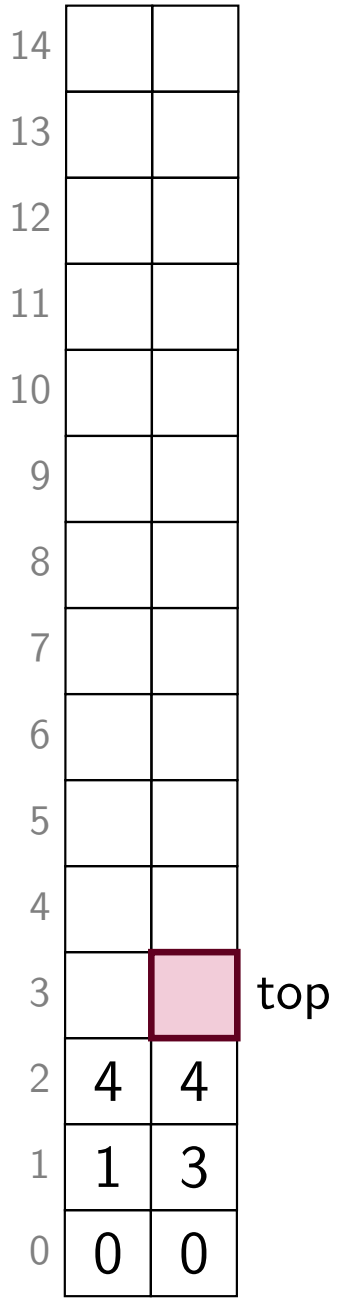
p a,b

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

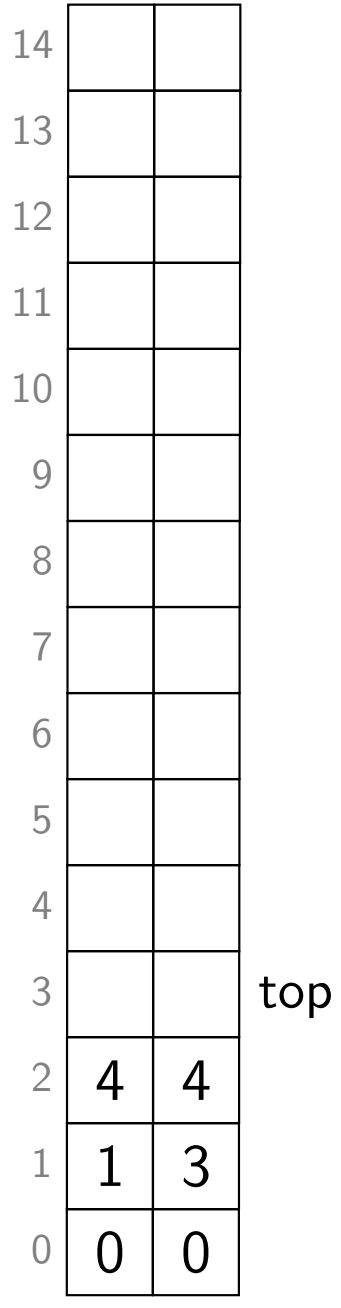


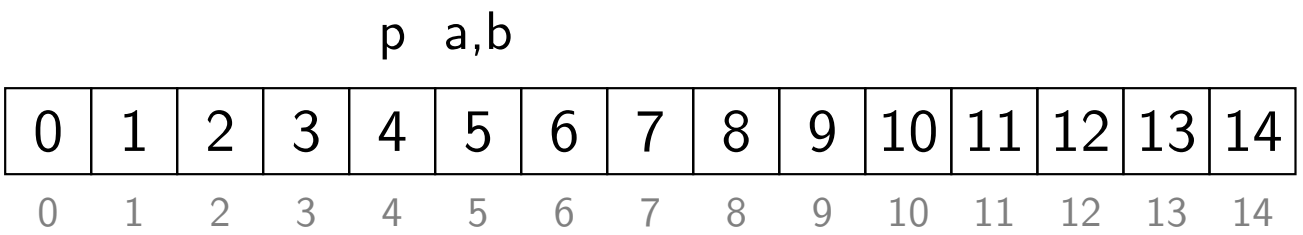
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

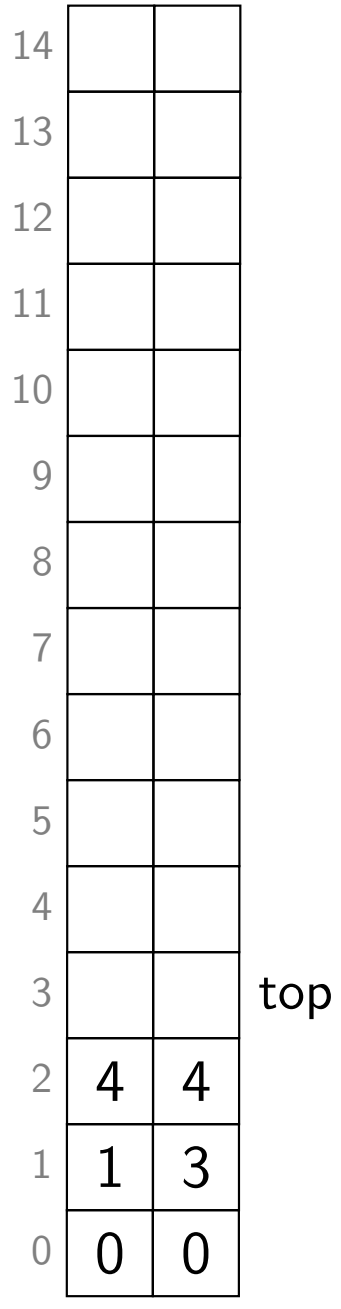
p a,b

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

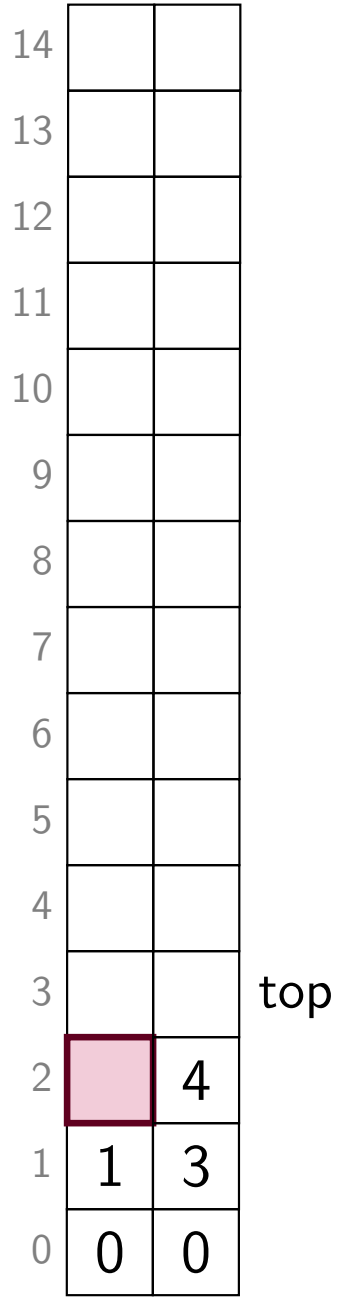


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

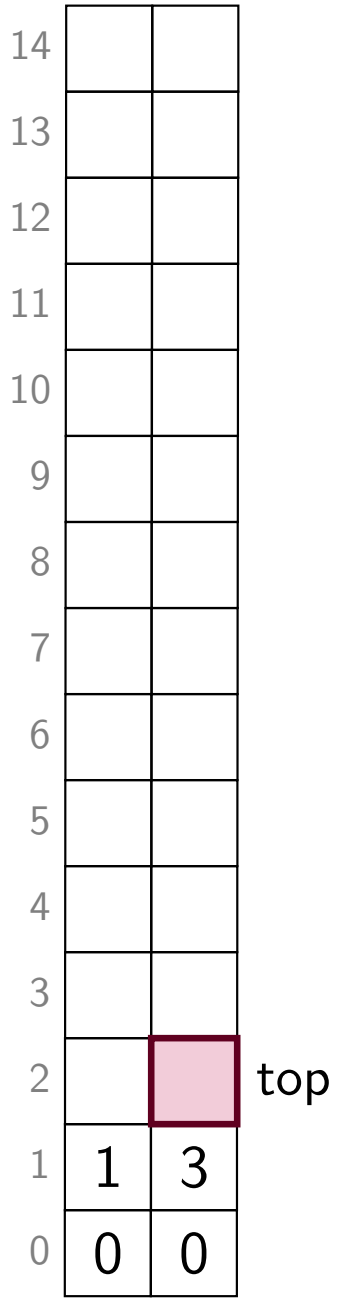
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

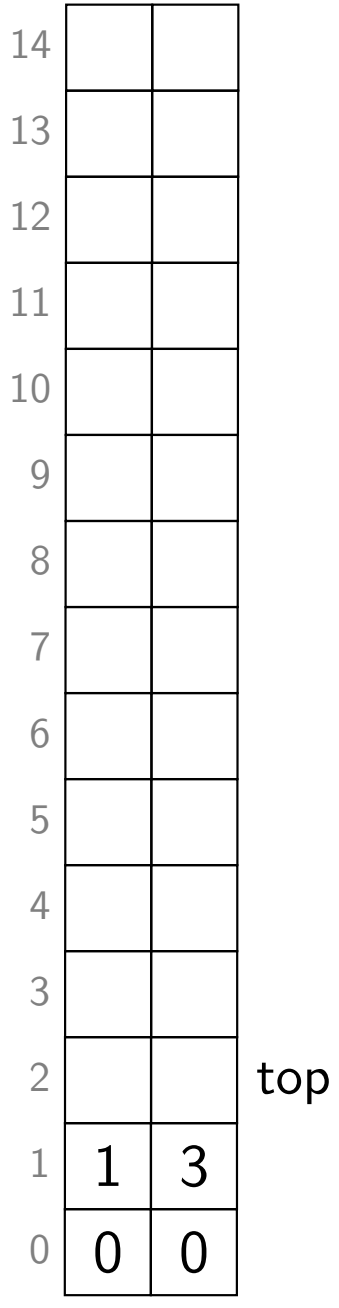
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

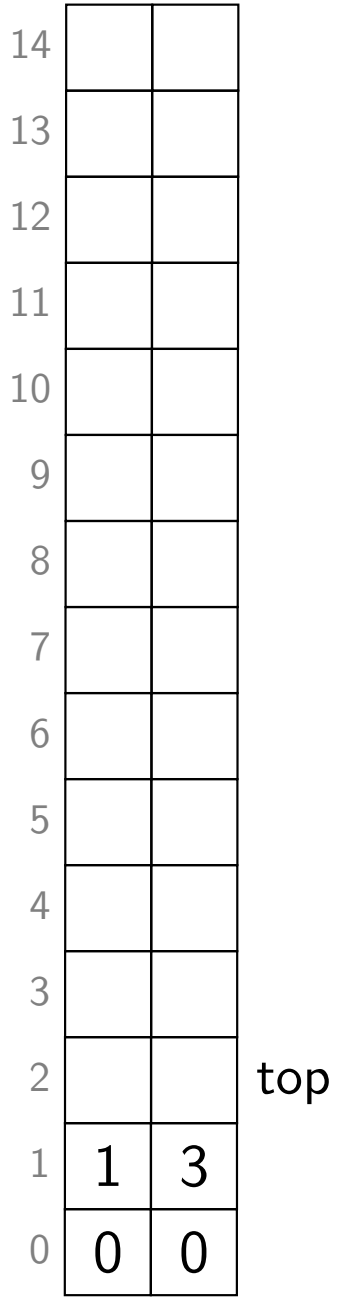
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



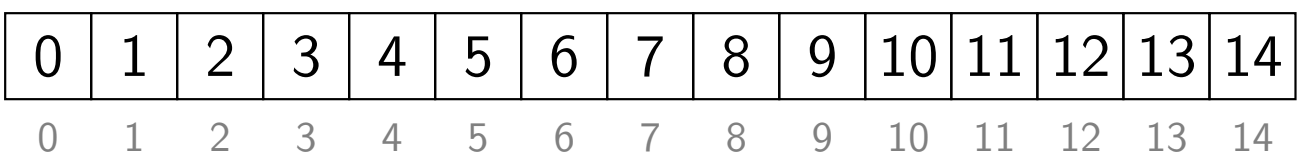
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

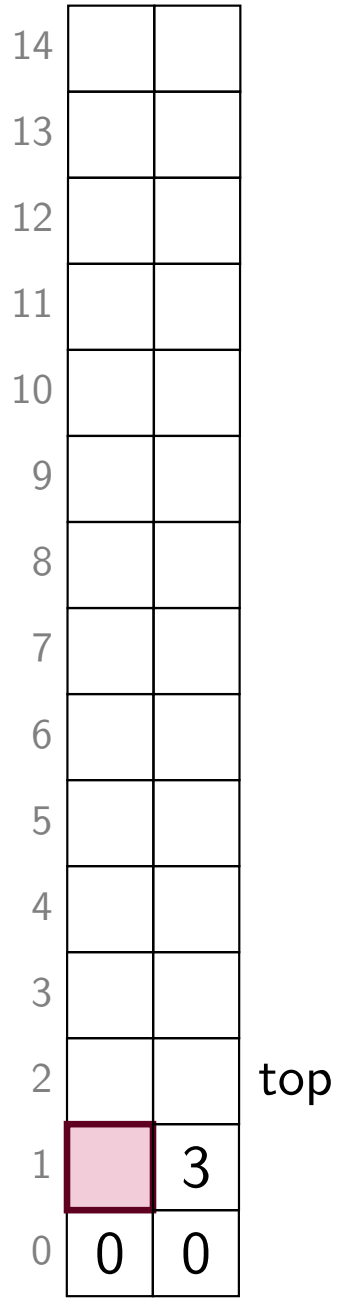
a,b,p

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

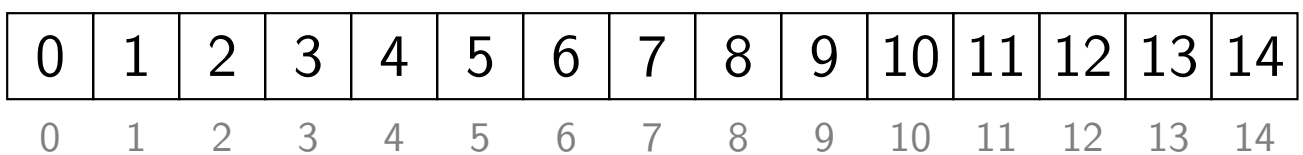


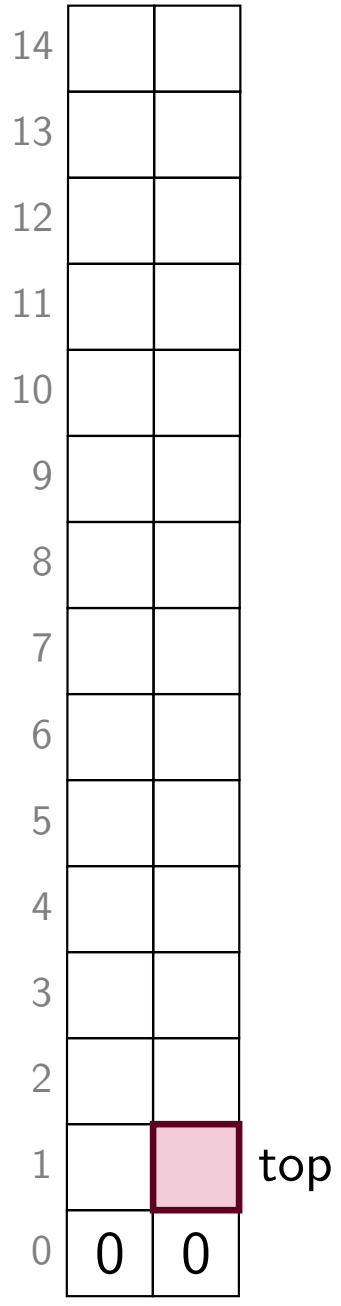
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



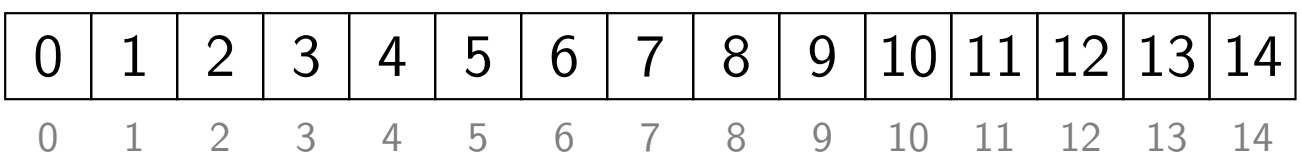



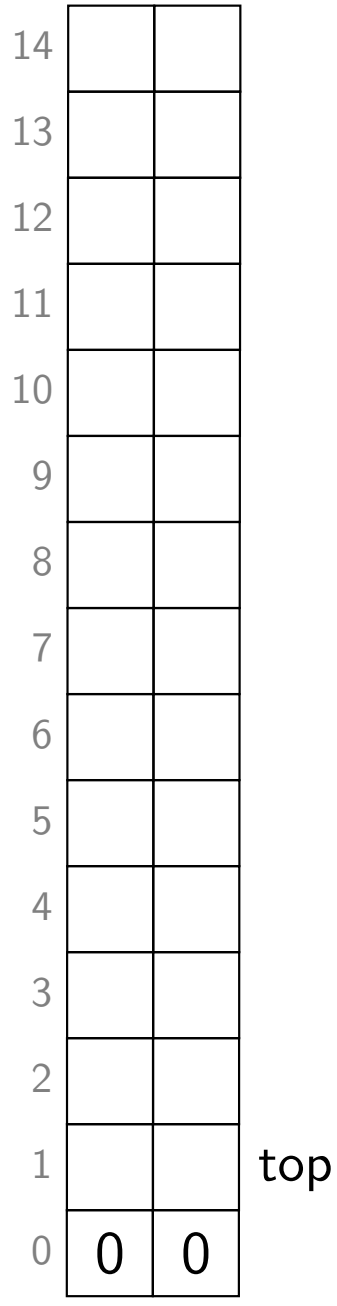
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



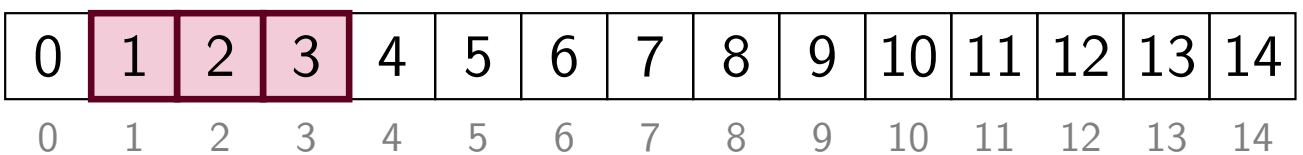



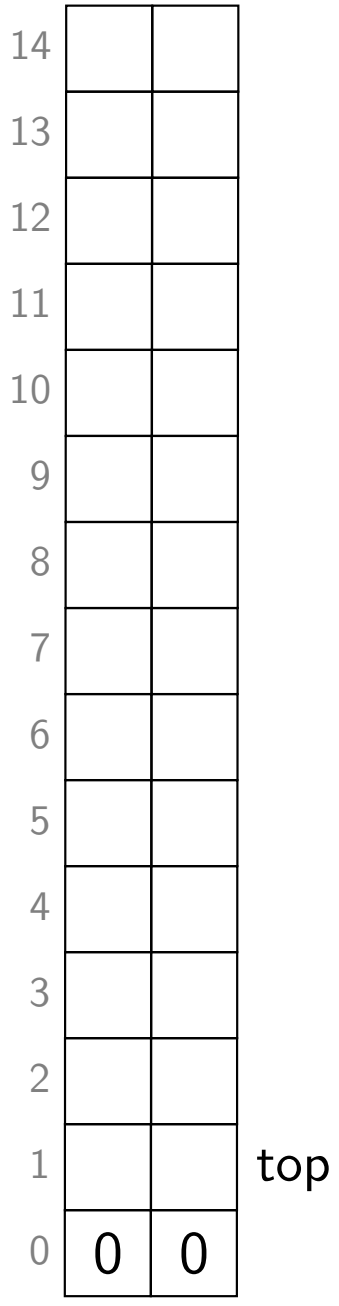
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



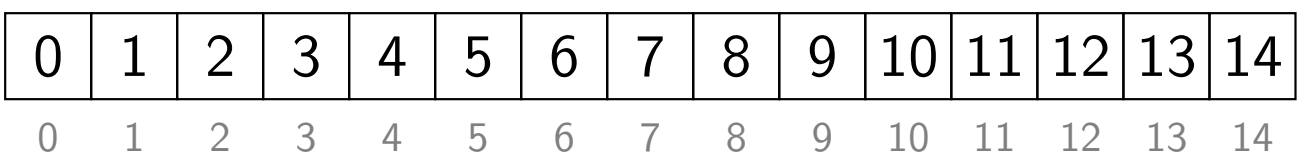



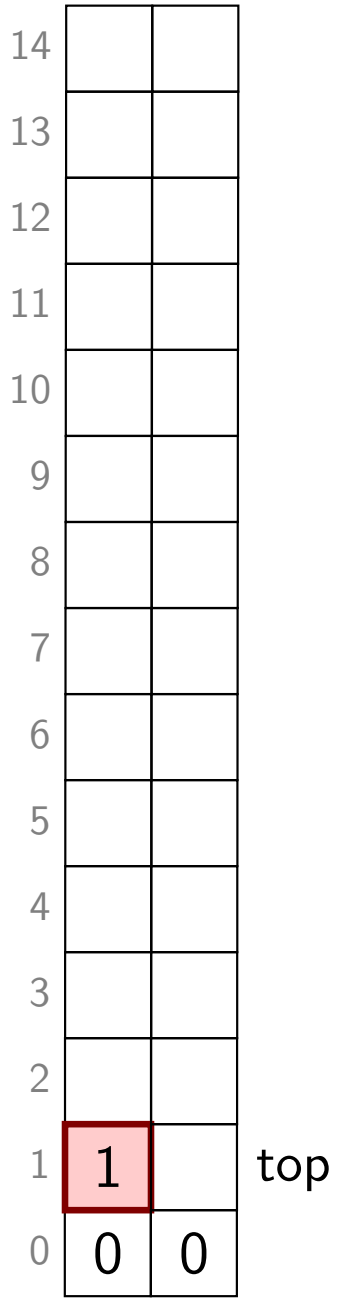
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

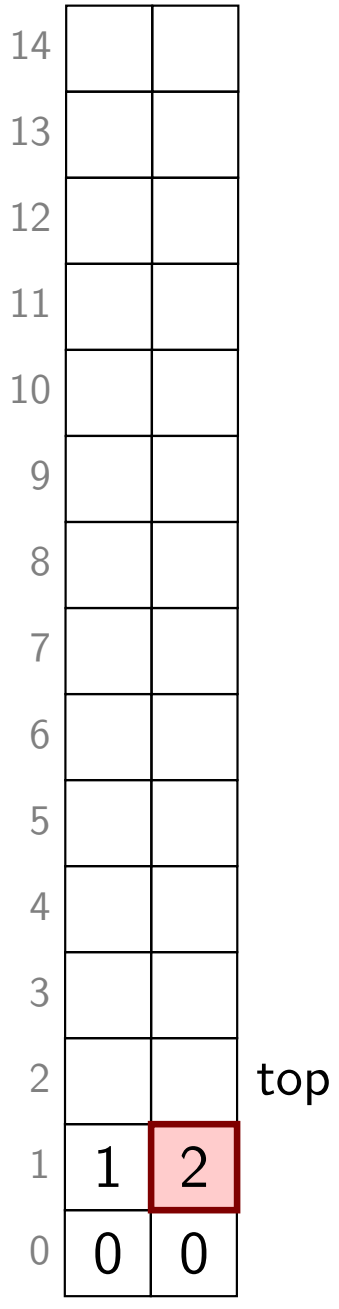


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a p b

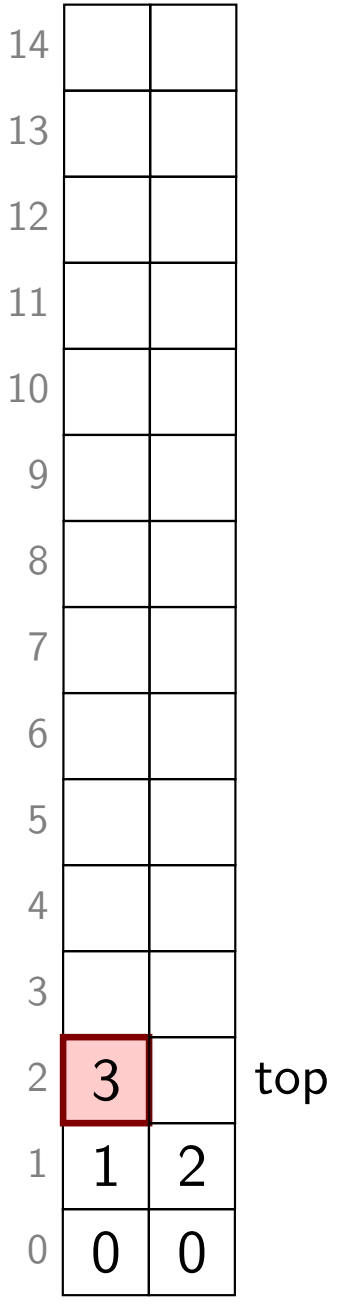
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



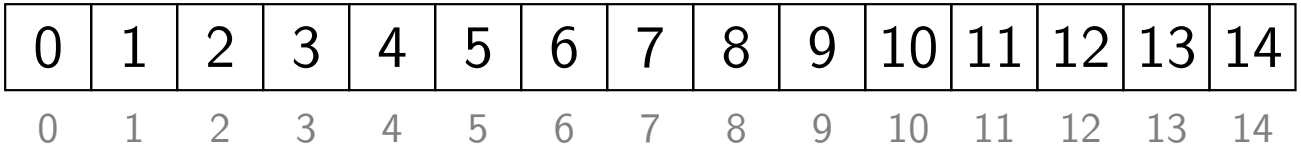
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

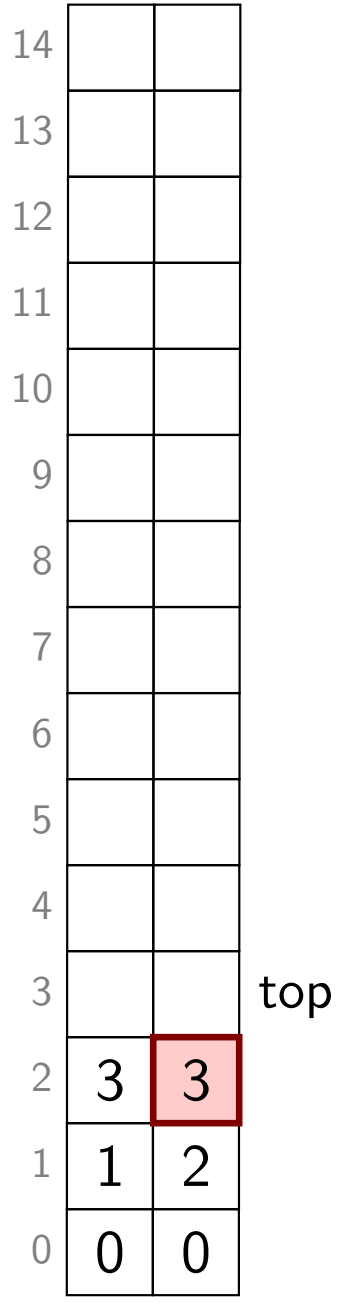
a p b

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

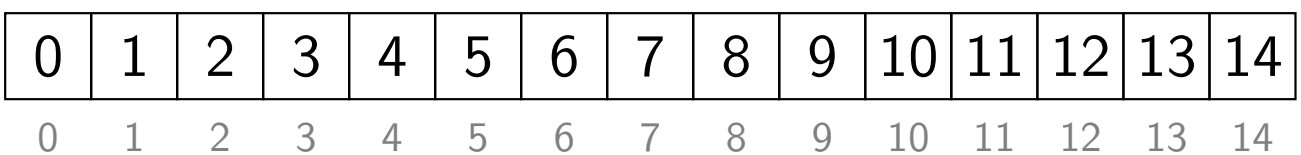


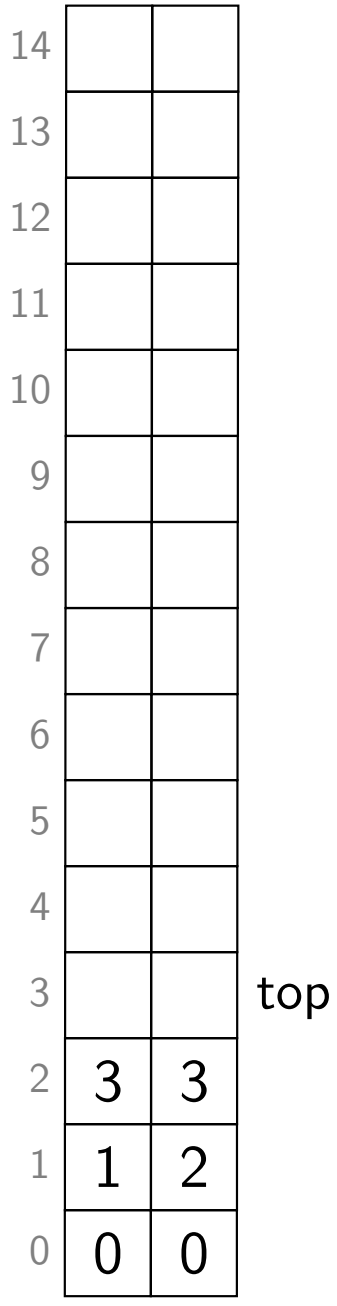
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

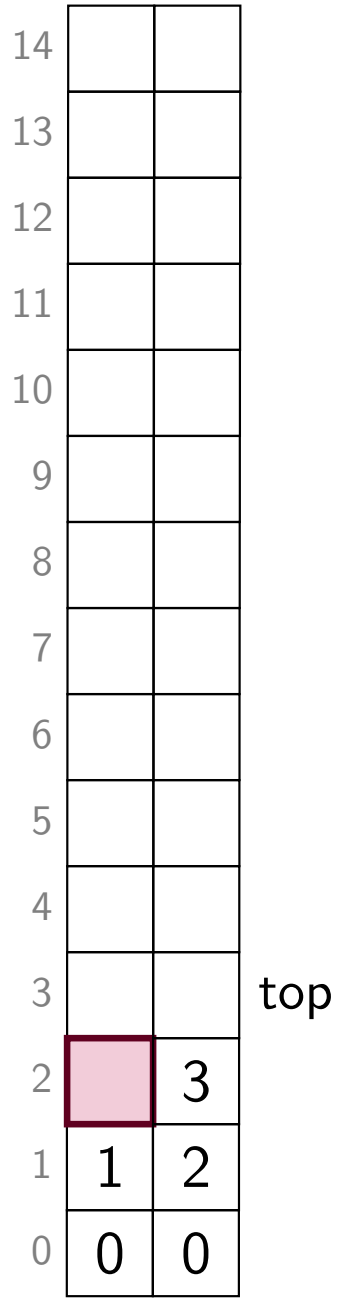


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

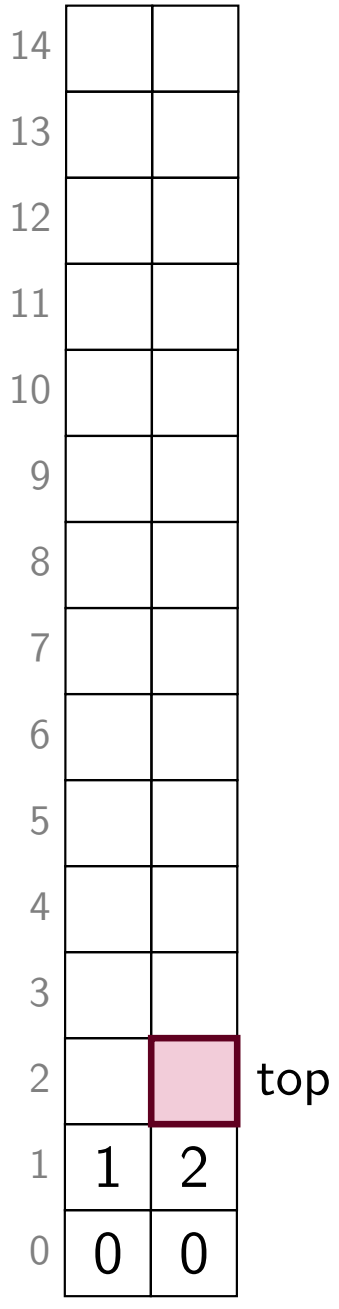
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

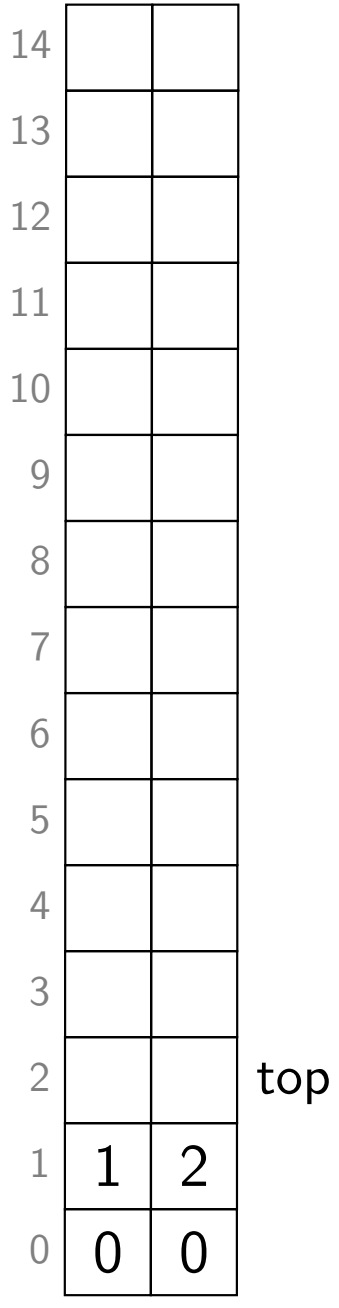
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



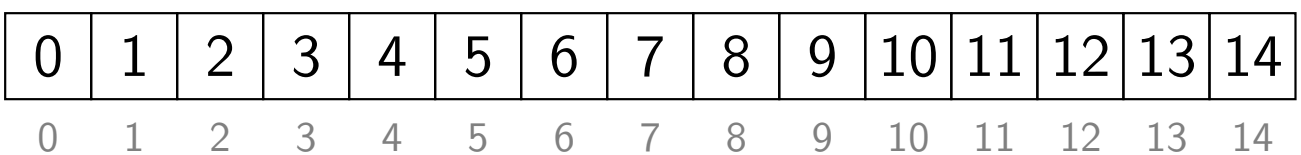
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

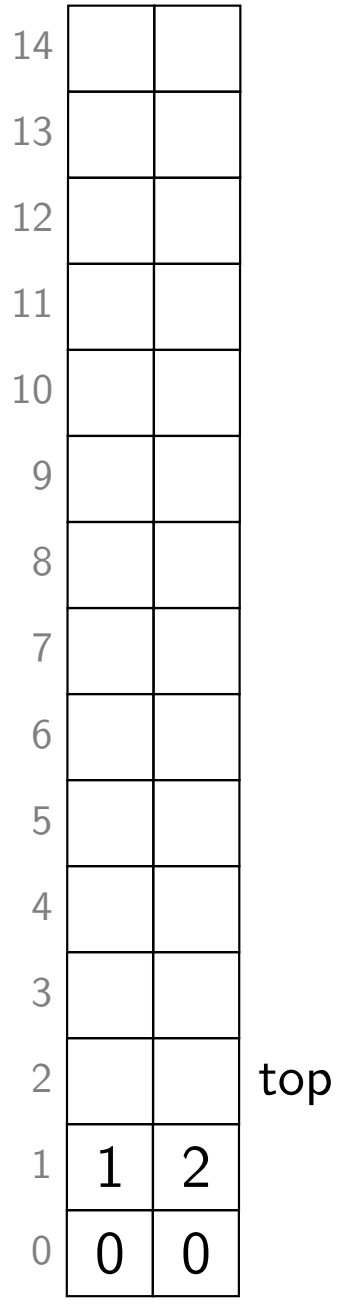
p a,b

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

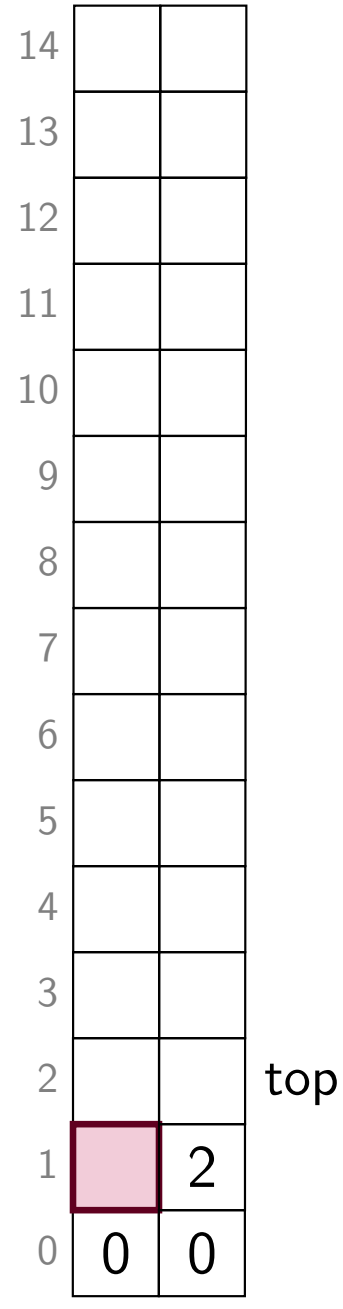


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a p b

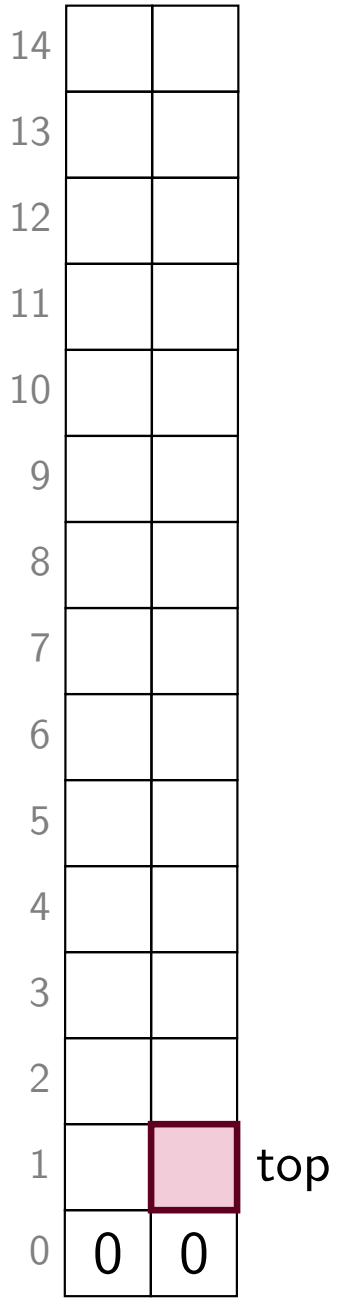
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a b,p

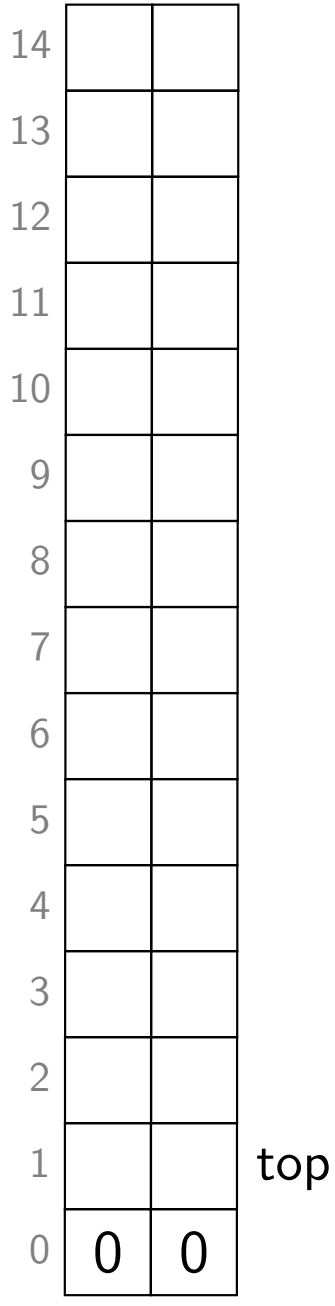
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



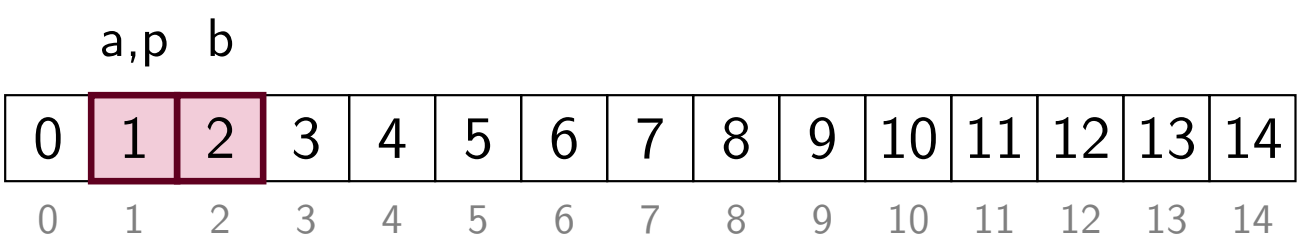
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

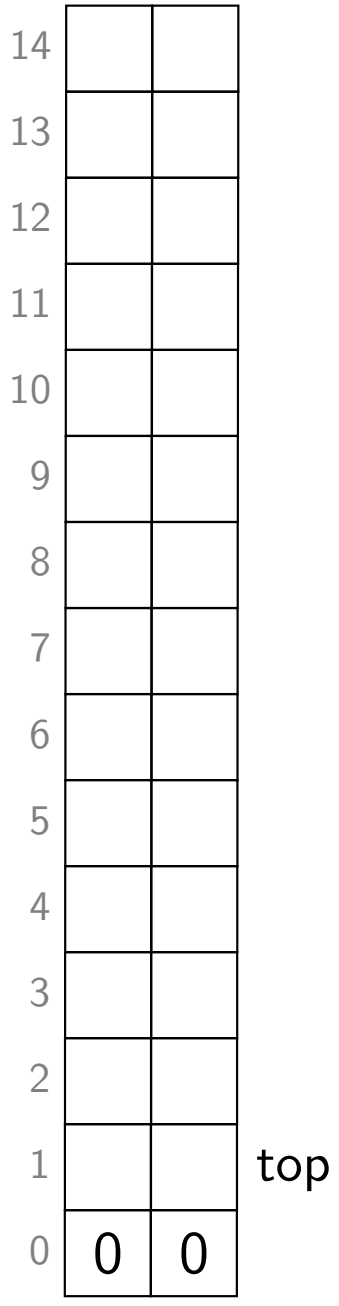
a b,p

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

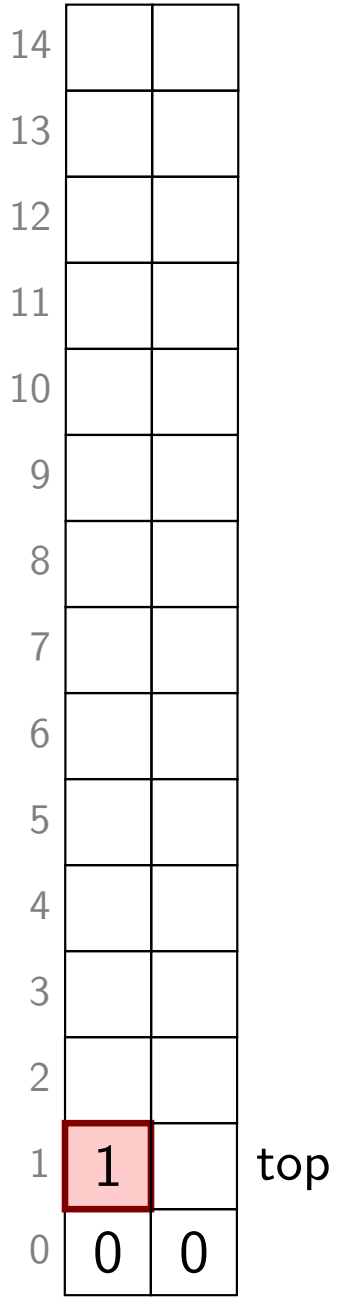


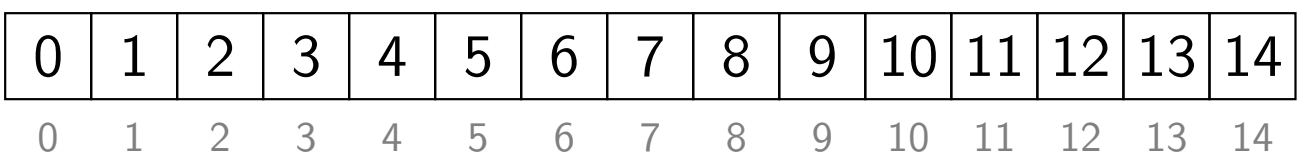
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

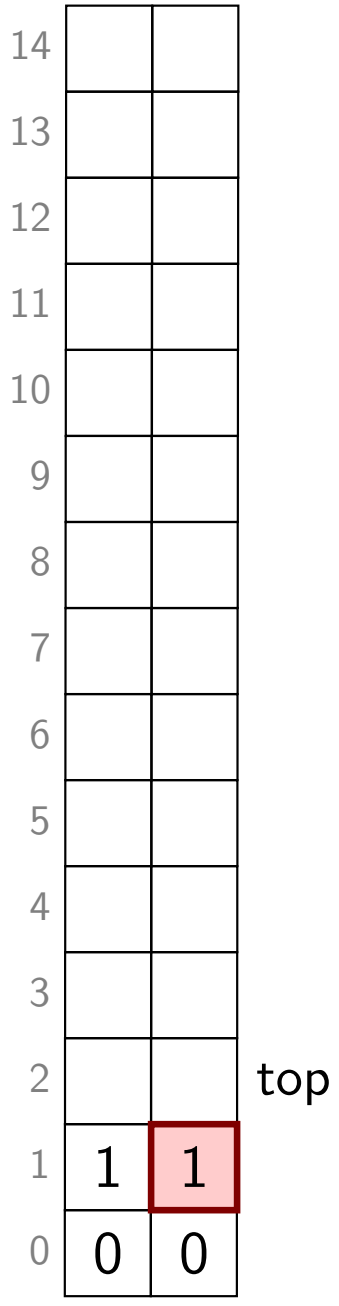
a,p b

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

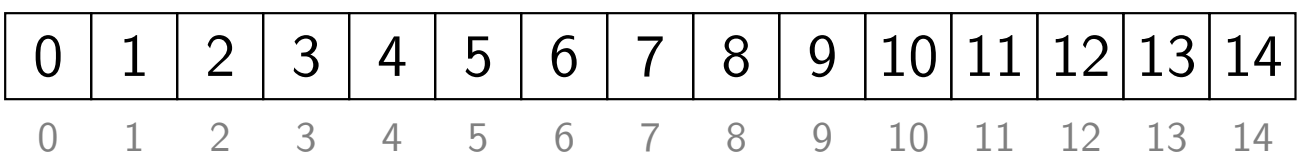


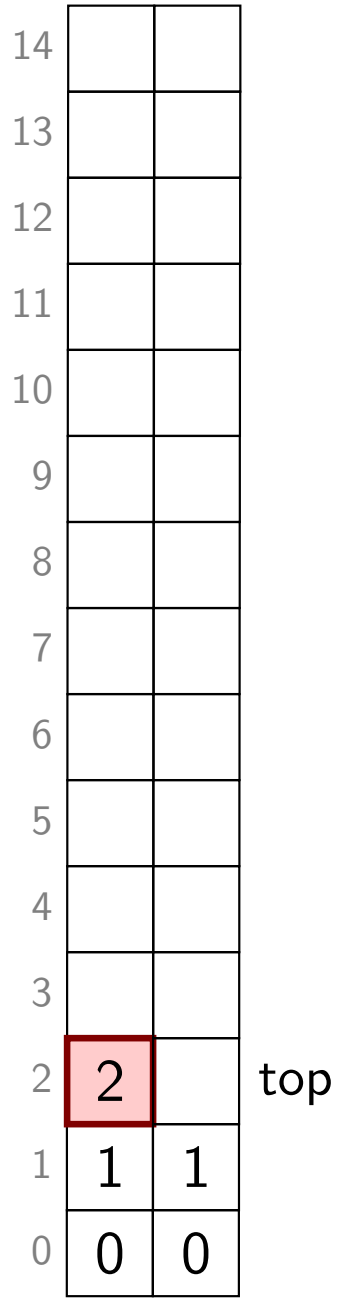
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



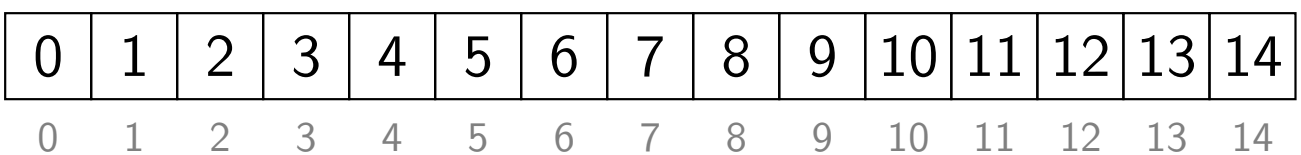



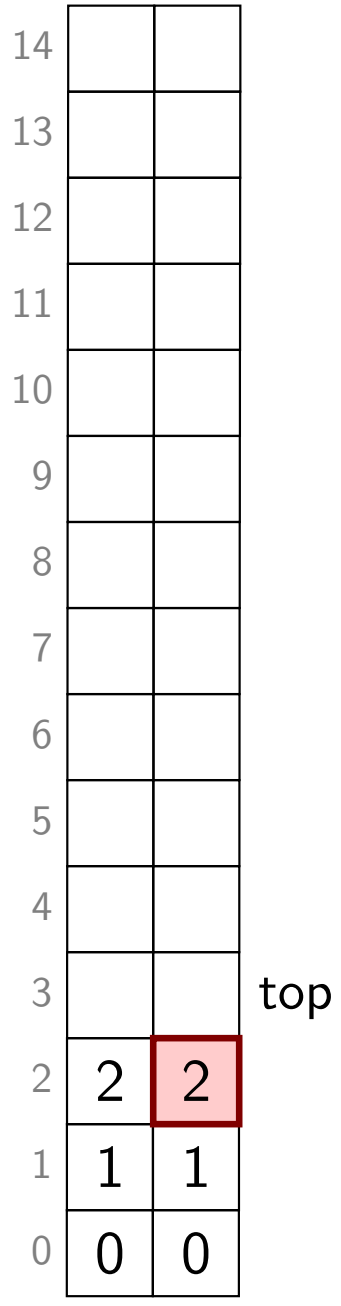
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



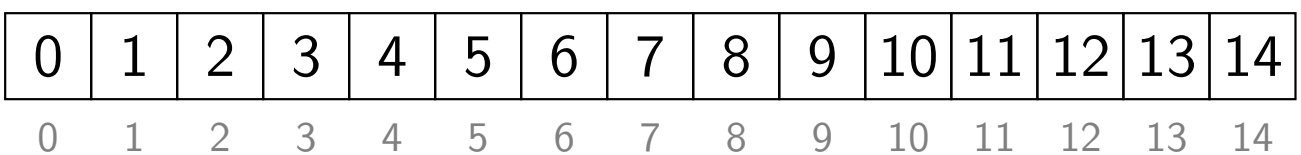



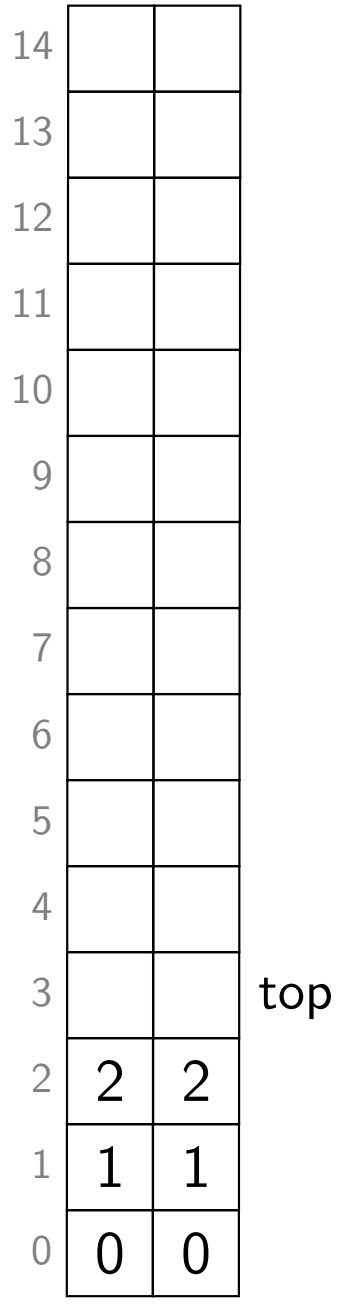
```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

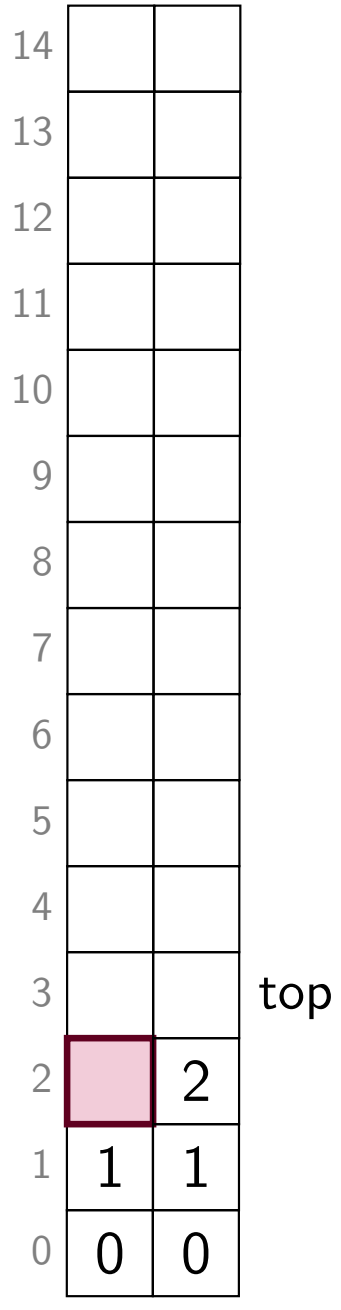


```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

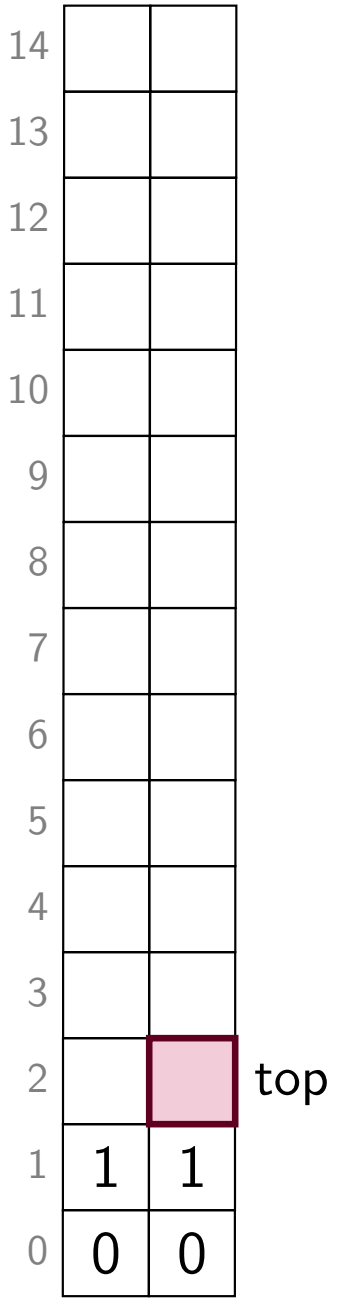
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

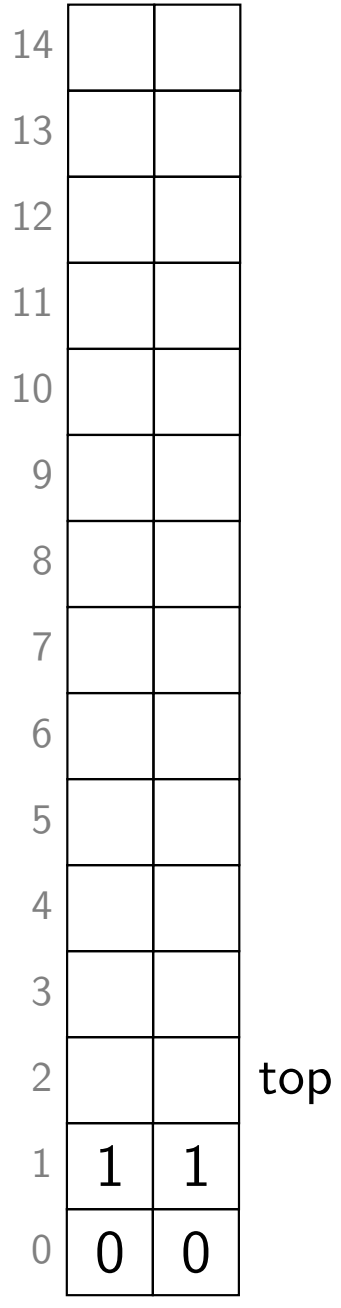
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

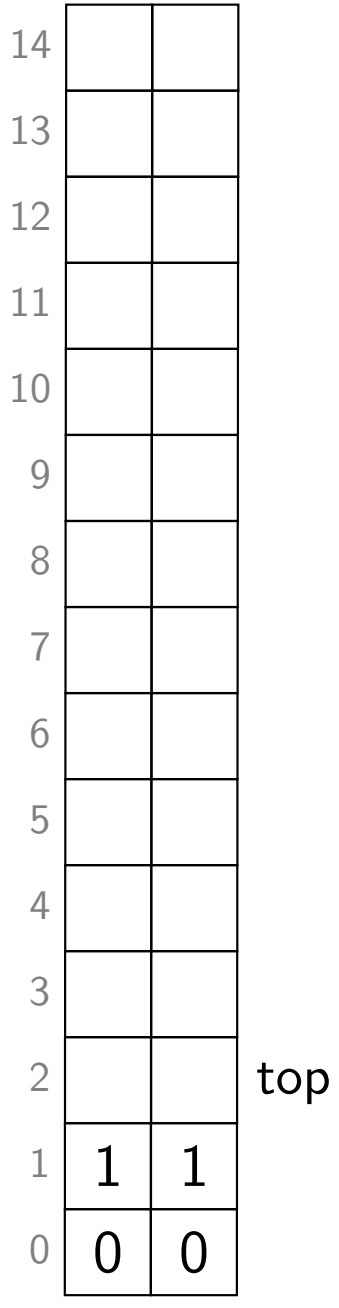
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

p a,b

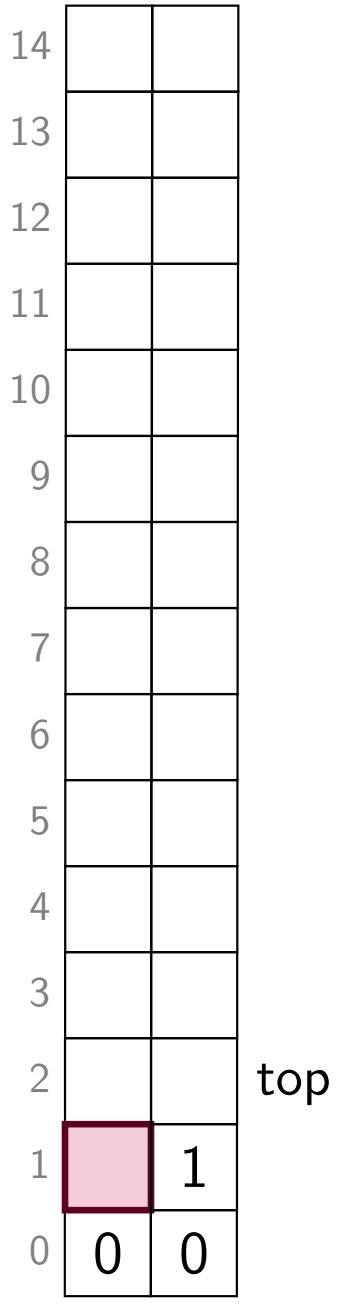
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,p b

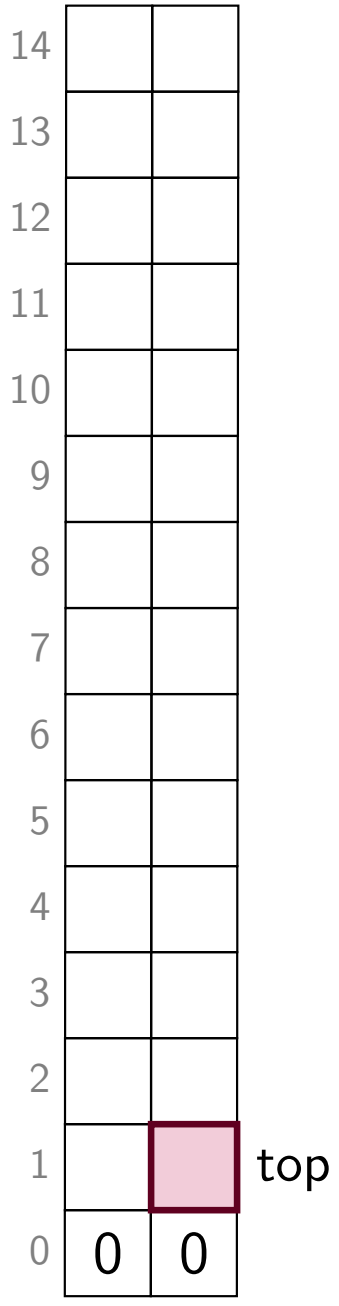
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

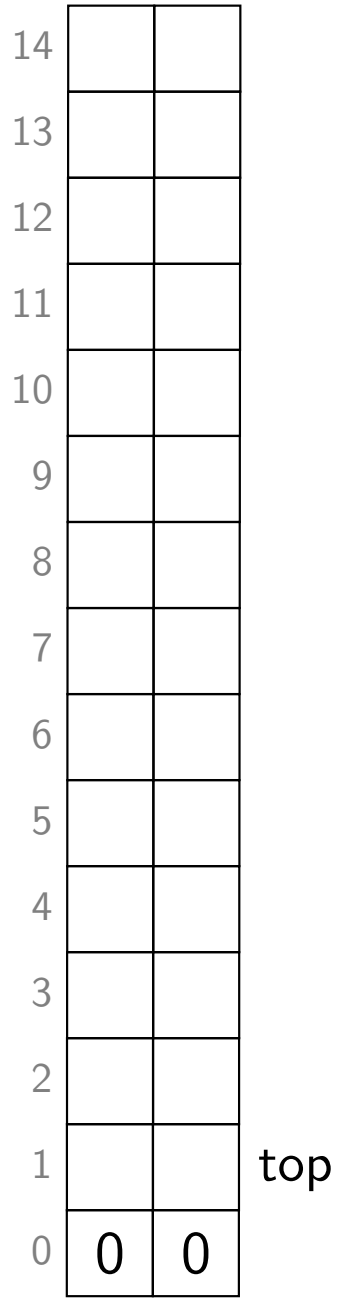
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

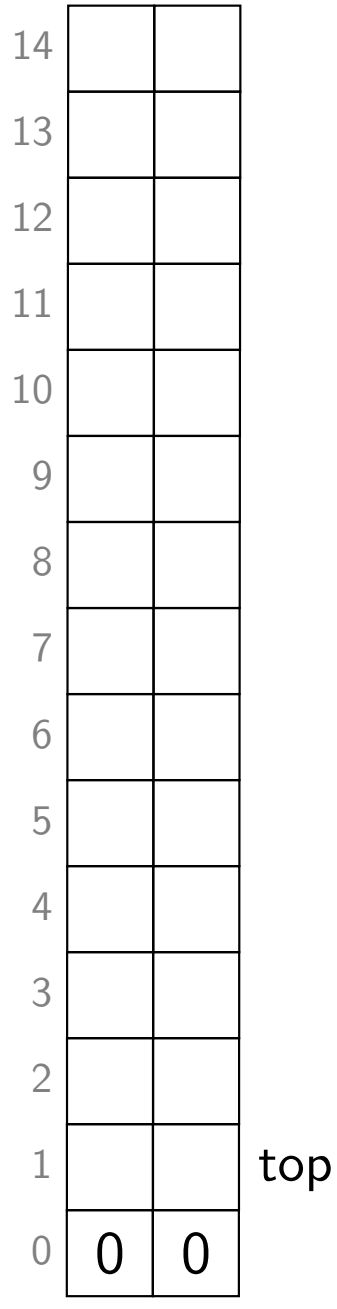
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b,p

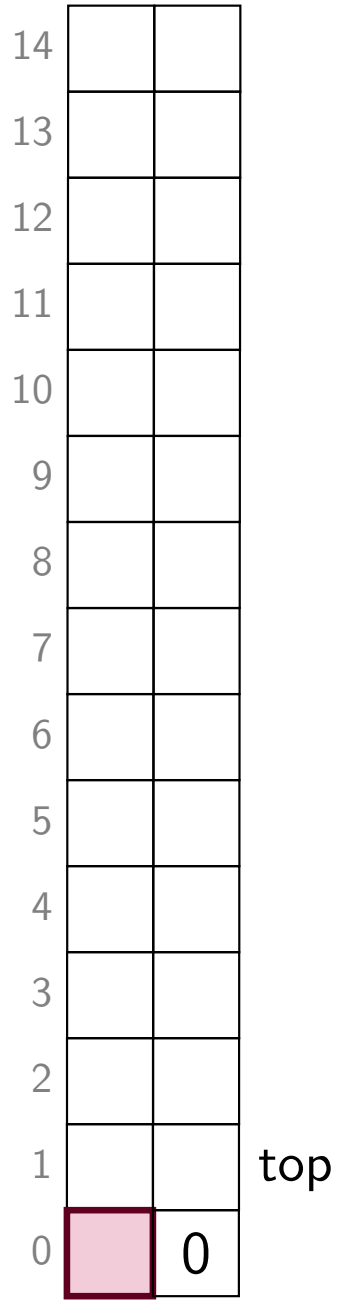
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a b,p

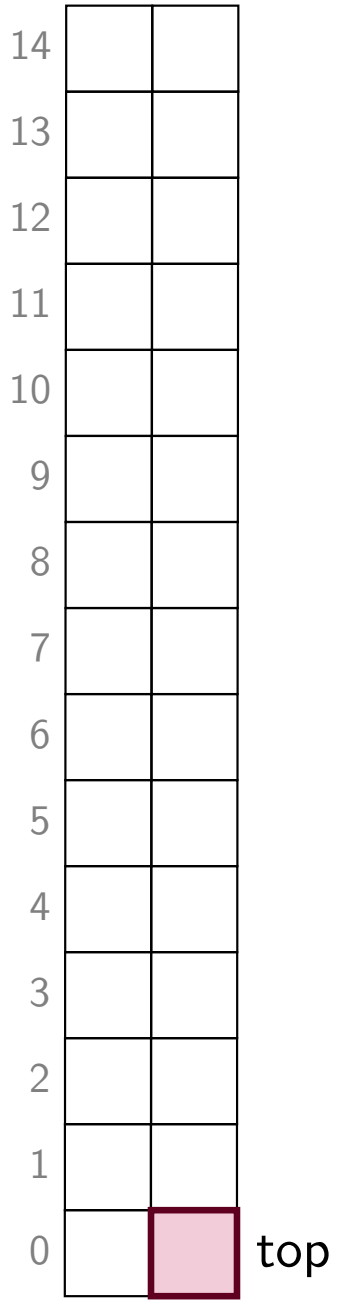
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

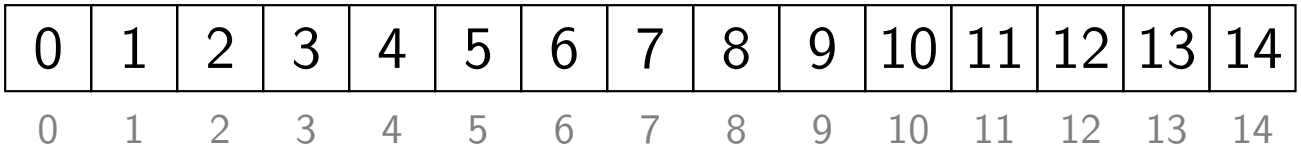
a,b p

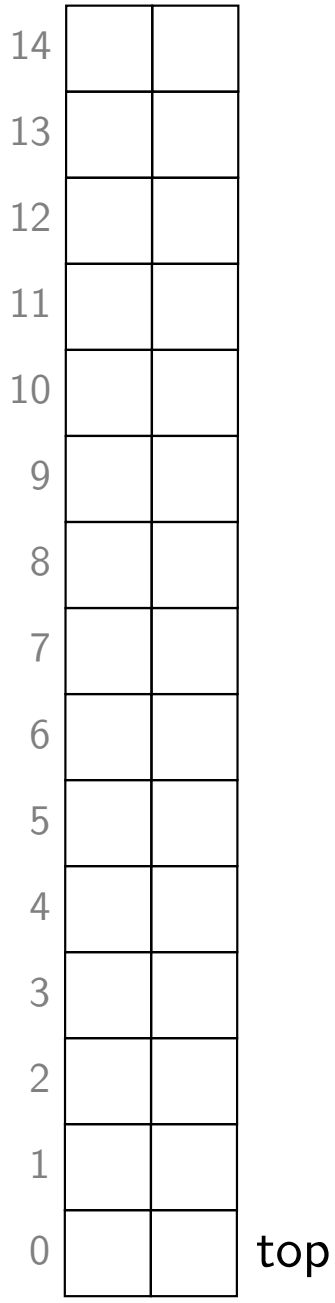
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b p



```
void sort (Tab t) {
  uint p, a, b;
  Stack first = stackNew(t.len);
  Stack last = stackNew(t.len);
  stackPush(&first, 0);
  stackPush(&last, t.len-1);
  while (!stackEmpty(first)) {
    a = stackPop(&first);
    b = stackPop(&last);
    if (a < b) {
      p = a + partition(slice(t, a, b));
      stackPush(&first, a);
      stackPush(&last, p);
      stackPush(&first, p+1);
      stackPush(&last, b);
    }
  }
}
```

a,b p