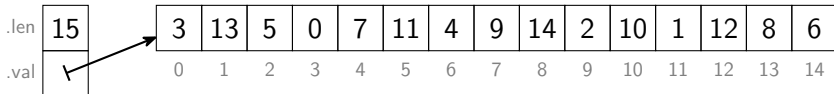```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];
  uint a = -1;
  uint b = t.len;
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

| .len | 15 | | 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|------|----|--|---|----|---|---|---|----|---|---|----|---|----|---|----|---|---|
| .val | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];
  uint a = -1;
  uint b = t.len;
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
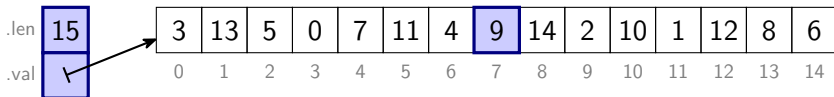
| .len | 15 |
|------|----|

| 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 |

.val
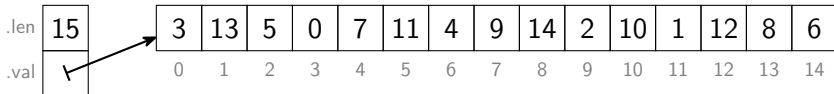
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1;
  uint b = t.len;
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
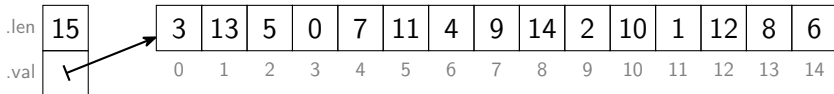
| .len | 15 | | 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
|------|----|----|---|----|---|---|---|----|---|---|----|---|----|---|----|---|---|
| .val | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;
  uint b = t.len;
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

.len	15		3	13	5	0	7	11	4	9	14	2	10	1	12	8	6
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* -1 */
  uint b = t.len;
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
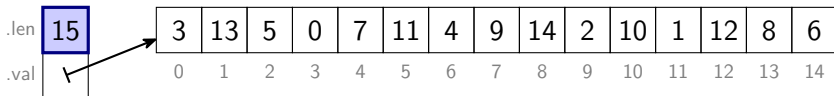
.len	15		3	13	5	0	7	11	4	9	14	2	10	1	12	8	6
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];   /* 9 */
  uint a = -1;   /* -1 */
  uint b = t.len;
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
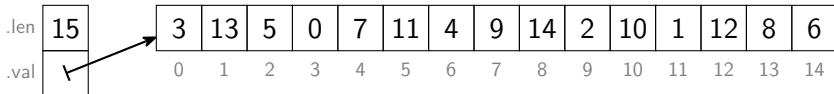
.len	15

.val		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

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];   /* 9 */
  uint a = -1;   /* -1 */
  uint b = t.len;   /* 15 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
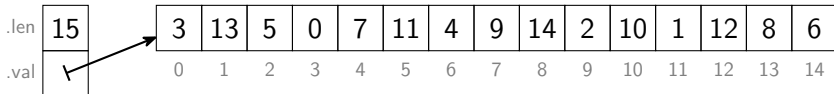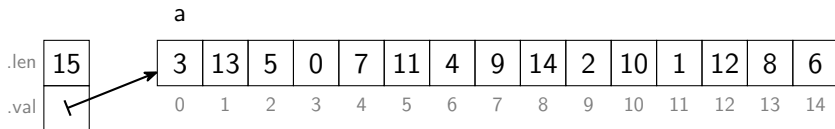

| .len | 15 | | 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
| .val | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

```c
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* -1 */
  uint b = t.len;  /* 15 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

.len	15		3	13	5	0	7	11	4	9	14	2	10	1	12	8	6
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 0 */
  uint b = t.len;  /* 15 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
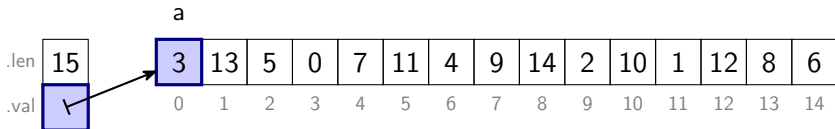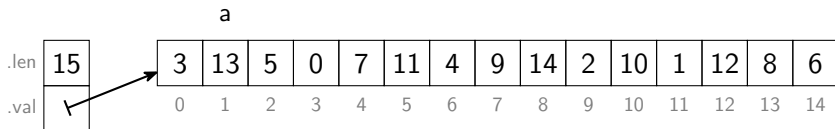
a

| .len | 15 | | 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
| .val | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 0 */
  uint b = t.len;  /* 15 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
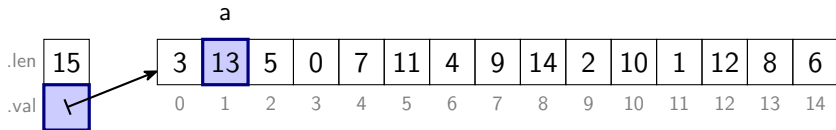
a

| .len | 15 | | 3 | 13 | 5 | 0 | 7 | 11 | 4 | 9 | 14 | 2 | 10 | 1 | 12 | 8 | 6 |
| .val | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11| 12 | 13| 14|

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 1 */
  uint b = t.len;  /* 15 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
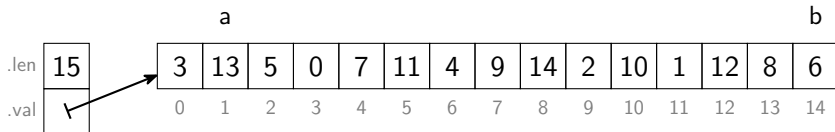
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 1 */
  uint b = t.len;  /* 15 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
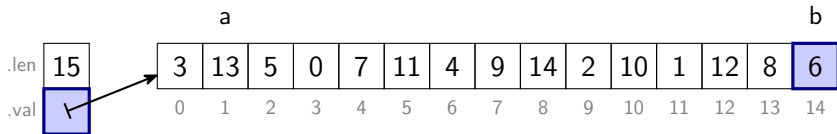
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 1 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
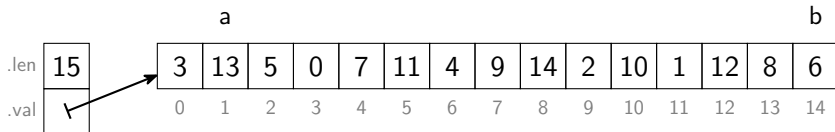
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 1 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 1 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
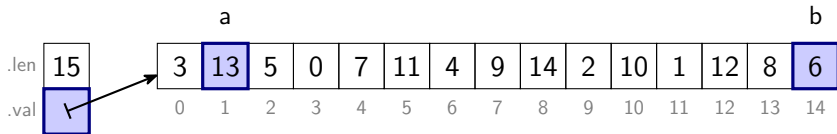
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1; /* 1 */
  uint b = t.len; /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```
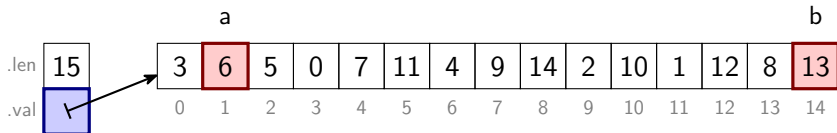
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 1 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```

```
uint partition (Tab t) {
  int pivot = t.val[(t.len−1)/2];  /* 9 */
  uint a = −1;  /* 1 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
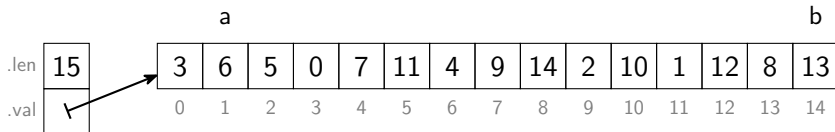
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 2 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
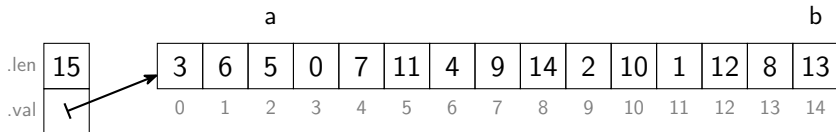
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1; /* 2 */
  uint b = t.len; /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 3 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
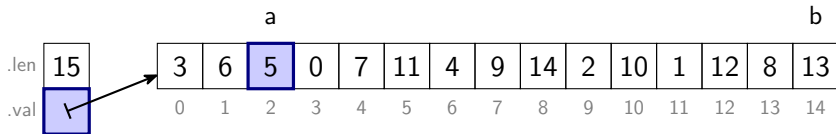
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 3 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
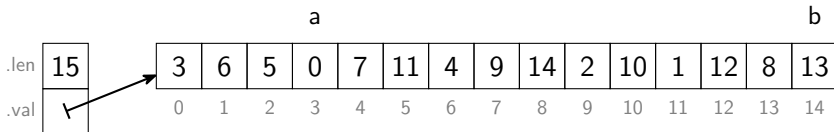
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 4 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
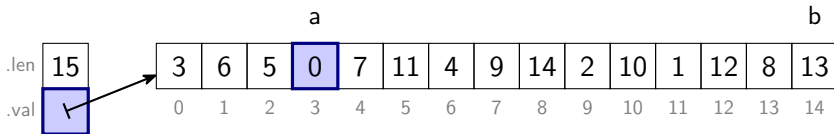
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];   /* 9 */
  uint a = -1;   /* 4 */
  uint b = t.len;   /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
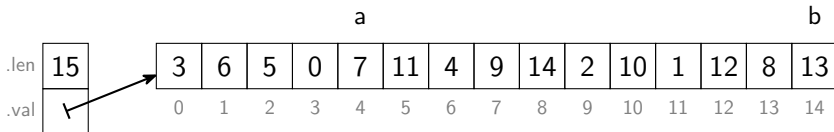
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 5 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 5 */
  uint b = t.len;  /* 14 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 5 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
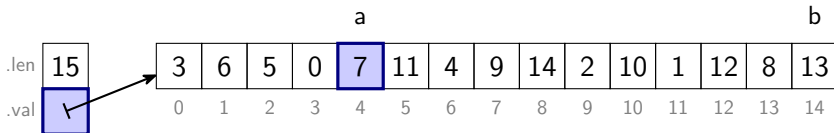
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 5 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

				a										b	
.len	15	3	6	5	0	7	11	4	9	14	2	10	1	12	8	13
.val		0	1	2	3	4	5	6	7	8	9	10| 11	12| 13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len−1)/2];  /* 9 */
  uint a = −1;  /* 5 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
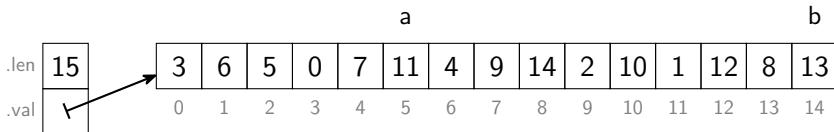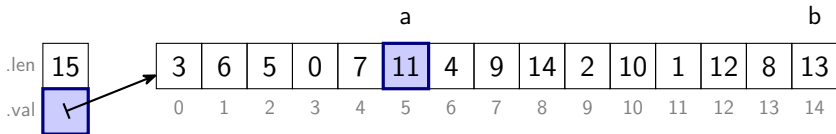
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1; /* 5 */
  uint b = t.len; /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```
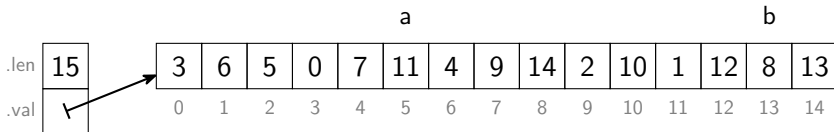
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 5 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```

.len	15		3	6	5	0	7	8	4	9	14	2	10	1	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

a (at index 5), b (at index 13)

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1; /* 5 */
  uint b = t.len; /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
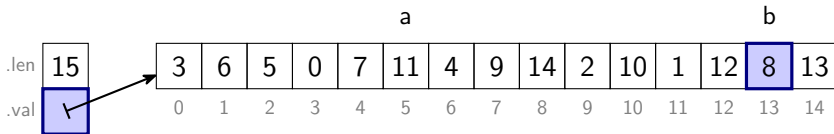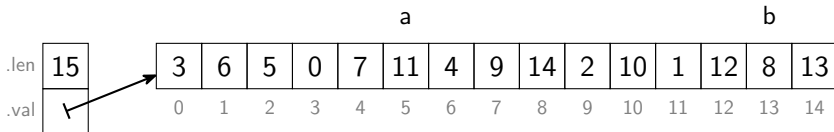
		a											b	

.len	15
.val	

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

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 6 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
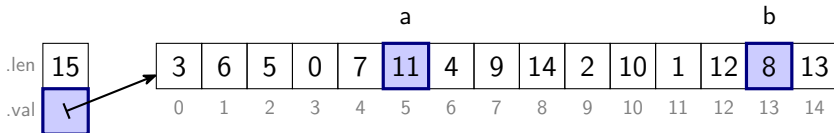
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 6 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
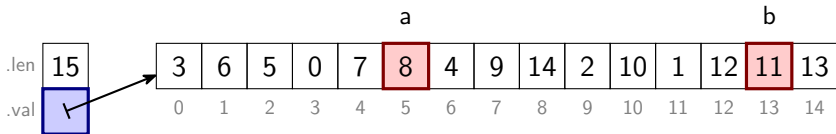
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

			a										b		
.len	15	3	6	5	0	7	8	4	9	14| 2	10| 1	12	11	13
.val		0	1	2	3	4	5	6	7	8	9	10| 11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 13 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
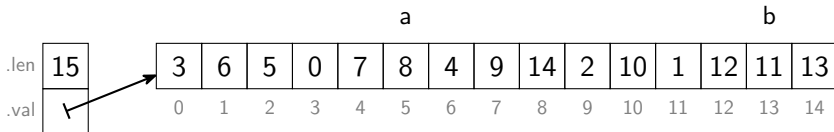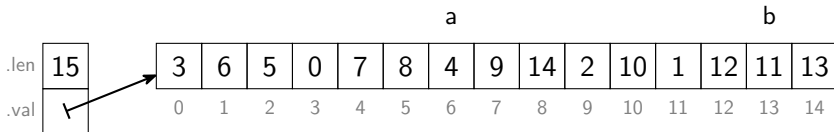
									a						b
.len	15		3	6	5	0	7	8	4	9	14	2	10	1	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 12 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
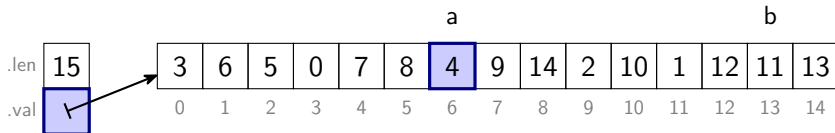
			a									b			
.len 15	3	6	5	0	7	8	4	9	14	2	10	1	12	11	13
.val	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 12 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
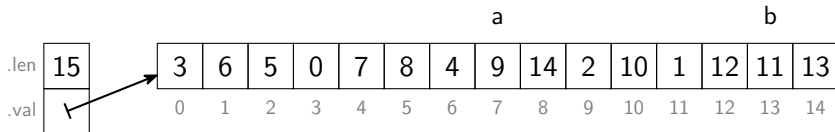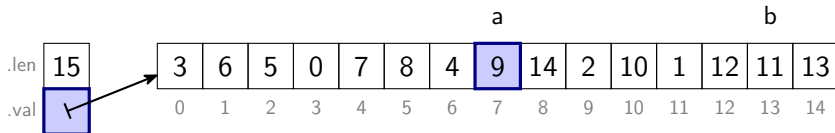
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

			a								b			
.len	15	3	6	5	0	7	8	4	9	14	2	10	1	12	11	13
.val		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];   /* 9 */
  uint a = -1;   /* 7 */
  uint b = t.len;   /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
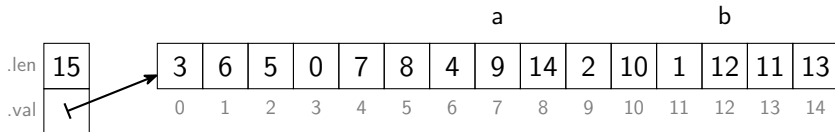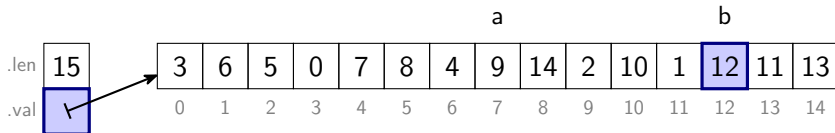
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

									a				b			
.len	15	3	6	5	0	7	8	4	9	14	2	10	1	12	11	13
.val		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 7 */
  uint b = t.len;  /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```
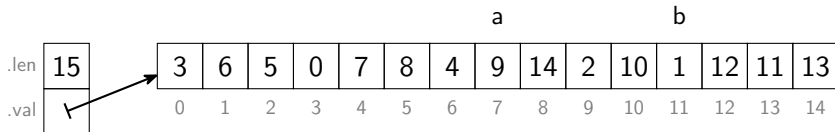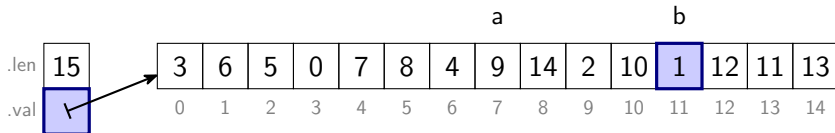
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];   /* 9 */
  uint a = -1;   /* 7 */
  uint b = t.len;   /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```
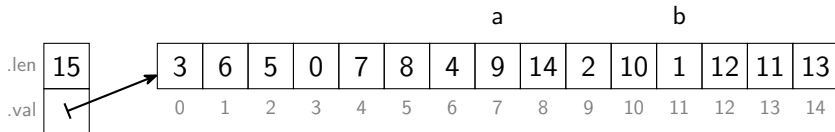
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];   /* 9 */
  uint a = -1;   /* 7 */
  uint b = t.len;   /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

		a								b				

.len	15

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

.val

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
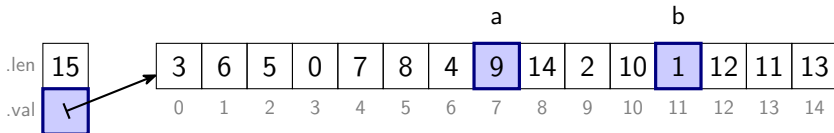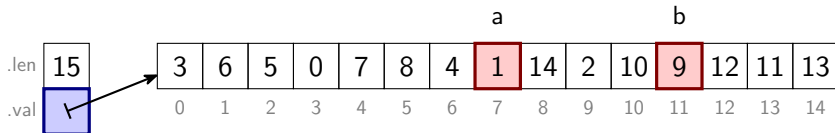
		a									b				

.len `15`

.val

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

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 11 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
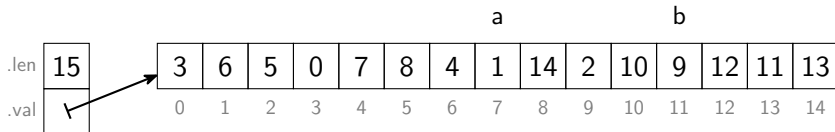
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 10 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
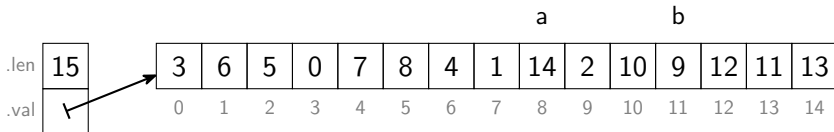
	a		b	

.len	15		3	6	5	0	7	8	4	1	14	2	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 10 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
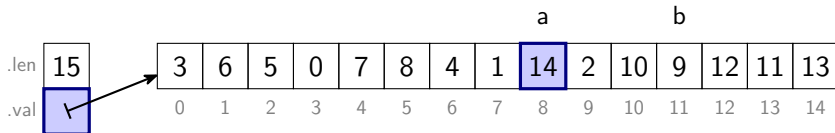
											a		b			
.len	15		3	6	5	0	7	8	4	1	14	2	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
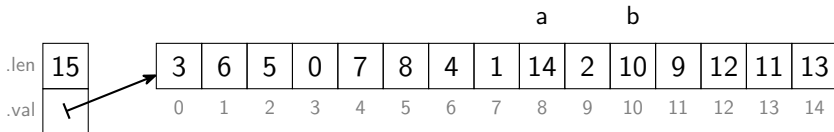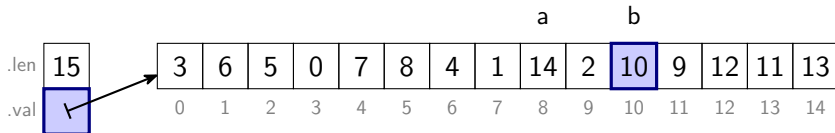
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
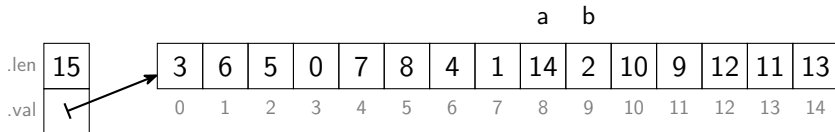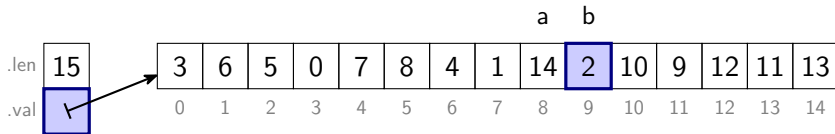
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

		a	b										

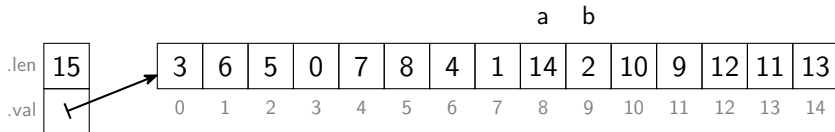	.len	15		3	6	5	0	7	8	4	1	14	2	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1; /* 8 */
  uint b = t.len; /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```

	a	b

.len	15		3	6	5	0	7	8	4	1	14	2	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
    swap(t, a, b);
  }
}
```
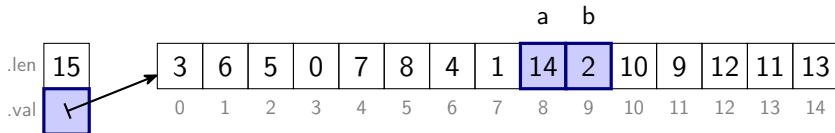
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 8 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

a   b

.len	15		3	6	5	0	7	8	4	1	2	14	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 9 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
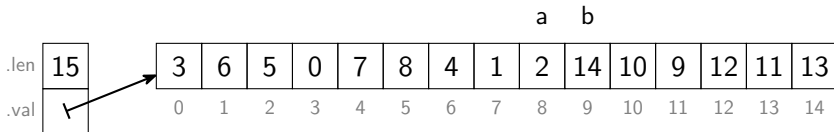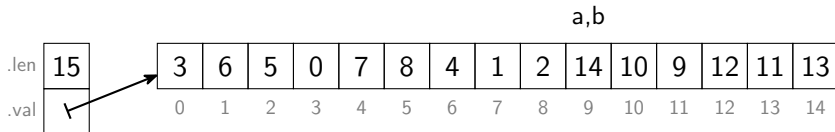
a,b

.len	15		3	6	5	0	7	8	4	1	2	14	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 9 */
  uint b = t.len;  /* 9 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
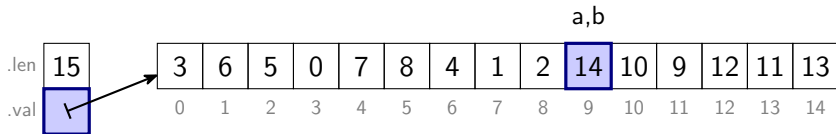
a,b

.len	15		3	6	5	0	7	8	4	1	2	14	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 9 */
  uint b = t.len;  /* 8 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
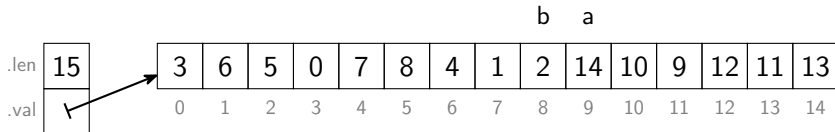

	b	a	

.len	15		3	6	5	0	7	8	4	1	2	14	10	9	12	11	13
.val			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 9 */
  uint b = t.len;  /* 8 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
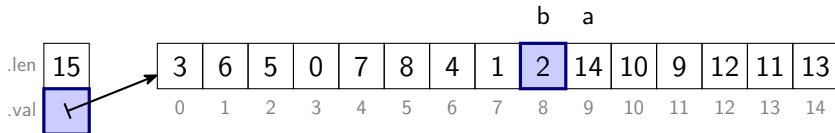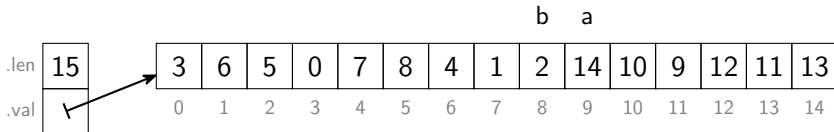
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2]; /* 9 */
  uint a = -1; /* 9 */
  uint b = t.len; /* 8 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```

	b	a												

.len	15
.val	

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

```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 9 */
  uint b = t.len;  /* 8 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```
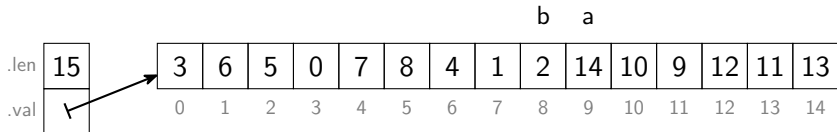
```
uint partition (Tab t) {
  int pivot = t.val[(t.len-1)/2];  /* 9 */
  uint a = -1;  /* 9 */
  uint b = t.len;  /* 8 */
  while (1) {
    do { a++; } while (t.val[a] < pivot);
    do { b--; } while (t.val[b] > pivot);
    if (a >= b) { return b; }
     swap(t, a, b);
  }
}
```