

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
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);
        }
    }
}

```

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

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

```

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

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

```

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

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

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

```

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

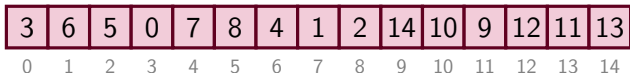
14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
0		

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);
        }
    }
}

```



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

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);
        }
    }
}

```

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

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

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1	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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1	9	14
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1	9	14
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1	14	
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
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);
        }
    }
}

```

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

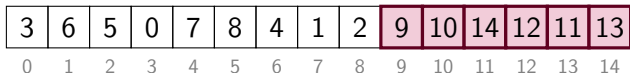
14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
0	0	8

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);
        }
    }
}

```



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

Diagram illustrating the stack state. The stack is represented by a vertical array of cells indexed 0 to 14. The bottom cell (index 0) contains the value 0, and the cell at index 1 contains the value 8. The label 'top' is placed next to the stack structure.

```

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	6	5	0	7	8	4	1	2	9	10	14	12	11	13
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		
1	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);
        }
    }
}

```

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

```

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

```

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

```

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

14			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2	10	14	
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);
        }
    }
}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

14			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3	13	14	
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);
        }
    }
}

```

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

```

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

```

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

```

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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
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);
        }
    }
}

```

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	
2	10	12
1	9	9
0	0	8

top

a, p 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	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	13
2	10	12
1	9	9
0	0	8

top

a, p 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	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	
3	13	13
2	10	12
1	9	9
0	0	8

top

a, p 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	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

a, p 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	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			top
4	14	14	
3	13	13	
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);
        }
    }
}

```

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	
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);
        }
    }
}

```

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	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);
        }
    }
}

```

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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3	13	13	
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);
        }
    }
}

```

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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3	13	13	
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);
        }
    }
}

```

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);
        }
    }
}

```

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	12
1	9	9
0	0	8

top

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);
        }
    }
}

```

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	12	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);
        }
    }
}

```

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	12	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);
        }
    }
}

```

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	12		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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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	
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);
        }
    }
}

```

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	11	
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);
        }
    }
}

```

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3	12	
2	10	11
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);
        }
    }
}

```

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	12	12
2	10	11
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);
        }
    }
}

```

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	12		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);
        }
    }
}

```

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	11
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);
        }
    }
}

```

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	11	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);
        }
    }
}

```

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	11	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);
        }
    }
}

```

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	11		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 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		
2		
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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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	
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);
        }
    }
}

```

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	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);
        }
    }
}

```

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	11	
2	10	10
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);
        }
    }
}

```

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	11	11
2	10	10
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);
        }
    }
}

```

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	11	11	
2	10	10	
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);
        }
    }
}

```

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	11		top
2	10	10	
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);
        }
    }
}

```

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	10
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);
        }
    }
}

```

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	10	
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);
        }
    }
}

```

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	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);
        }
    }
}

```

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,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		
2		
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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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		
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);
        }
    }
}

```

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		
1	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);
        }
    }
}

```

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		
2		
1		
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);
        }
    }
}

```

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		
1		
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);
        }
    }
}

```

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		
1		
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);
        }
    }
}

```

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		
1		
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);
        }
    }
}

```

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		
1		
0		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);
        }
    }
}

```

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		
1		
0		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);
        }
    }
}

```



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

```

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

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

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

```

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	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

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

```

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	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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

```

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

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

```

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

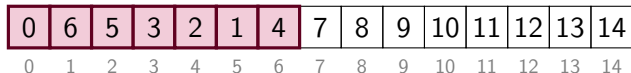
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

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

```



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

```

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

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

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

a
p
b

0	1	2	3	5	6	4	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		
1		
0	0	0

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);
        }
    }
}

```

a
p
b

0	1	2	3	5	6	4	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		
1	1	
0	0	0

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);
        }
    }
}

```

a
p
b

0	1	2	3	5	6	4	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		
1	1	3
0	0	0

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);
        }
    }
}

```

a
p
b

0	1	2	3	5	6	4	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	4	
1	1	3
0	0	0

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);
        }
    }
}

```

a                  p                  b

0	1	2	3	5	6	4	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	4	6
1	1	3
0	0	0

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);
        }
    }
}

```

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

14			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2	4	6	
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	6	4	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	6		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);
        }
    }
}

```

p   a   b

0	1	2	3	5	6	4	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		
1	1	3
0	0	0

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);
        }
    }
}

```

0	1	2	3	5	6	4	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		
1	1	3
0	0	0

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

14			
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			top
3	6	6	
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);
        }
    }
}

```

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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3	6	6	
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

14			
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3	6		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);
        }
    }
}

```

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	4	5
1	1	3
0	0	0

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);
        }
    }
}

```

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	4	5	
1	1	3	
0	0	0	

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);
        }
    }
}

```

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	4	5	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);
        }
    }
}

```

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

14			
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2	5		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);
        }
    }
}

```

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		
1	1	3
0	0	0

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);
        }
    }
}

```

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		
1	1	3
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1	1	3
0	0	0

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);
        }
    }
}

```

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		
3		
2	4	
1	1	3
0	0	0

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);
        }
    }
}

```

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			
3			
2	4	4	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);
        }
    }
}

```

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			
3	5		top
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);
        }
    }
}

```

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);
        }
    }
}

```

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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
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);
        }
    }
}

```

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			
3	5		top
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);
        }
    }
}

```

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		
3		
2	4	4
1	1	3
0	0	0

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);
        }
    }
}

```

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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
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);
        }
    }
}

```

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			
3			
2	4	4	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);
        }
    }
}

```

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			
3			
2	4		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);
        }
    }
    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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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

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

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);
        }
    }
}

```

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		
3		
2		
1	3	
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1	1	
0	0	0

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);
        }
        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		
3		
2		
1	1	2
0	0	0

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);
        }
        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		
3		
2	3	
1	1	2
0	0	0

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);
        }
        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		
3		
2	3	3
1	1	2
0	0	0

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);
        }
        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			top
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2	3	3	
1	1	2	
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

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

14		
13		
12		
11		
10		
9		
8		
7		
6		
5		
4		
3		
2		top
1	1	2
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);
        }
    }
}

```

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		
3		
2		
1	1	2
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1	1	2
0	0	0

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);
        }
    }
}

```

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

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1	1	
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1	1	1
0	0	0

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);
        }
    }
}

```

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		
3		
2	2	
1	1	1
0	0	0

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);
        }
    }
}

```

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		
3		
2	2	2
1	1	1
0	0	0

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);
        }
    }
}

```

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

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

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

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);
        }
        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

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

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);
        }
        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

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

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);
        }
    }
    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		
3		
2		
1	1	1
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	0

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);
        }
    }
}

```

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		
3		
2		
1		
0	0	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);
        }
    }
}

```

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		
3		
2		
1		
0		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);
        }
    }
}

```

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

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);
        }
    }
}

```

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		
3		
2		
1		
0		top