第四次作业

2.3.1

0 1 2 3 4 5 6 7 8 9 10 11
E A S Y Q U E S T I O N
E A S Y Q U E S T I O N
E A E Y Q U S S T I O N
E A E Y Q U S S T I O N
E A E Y Q U S S T I O N
E A E Y Q U S S T I O N

2.3.2

0 1 2 3 4 5 6 7 8 9 10 11
E A S Y Q U E S T I O N
E A E Y Q U S S T I O N
A E E Y Q U S S T I O N
A E E Y Q U S S T I O N
A E E N Q U S S T I O Y
A E E I N U S S T Q O Y
A E E I N U S S T Q O Y
A E E I N O S S T Q U Y
A E E I N O S S T Q U Y
A E E I N O Q S T S U Y
A E E I N O Q S T S U Y
A E E I N O Q S S T U Y
A E E I N O Q S S T U Y

2.3.3

floor(N / 2)

也就是向下取整

2.3.4

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
[20, 18, 16, 14, 12, 10, 8, 6, 4, 2]
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

2.3.5

```
 private static void sort(Comparable[] array) {

         int lt = 0;
         int gt = array.length - 1;
         int i = lt + 1;

         Comparable pivot = array[0];

         while (i <= gt) {
             int comparison = array[i].compareTo(pivot);

             if (comparison < 0) {
                 exchange(array, lt, i);
                 lt++;
                 i++;
             } else if (comparison > 0) {
                 exchange(array, i, gt);
                 gt--;
             } else {
                 i++;
             }
         }
     }

     private static void exchange(Comparable[] array, int position1, int position2) {
         Comparable temp = array[position1];
         array[position1] = array[position2];
         array[position2] = temp;
     }
```