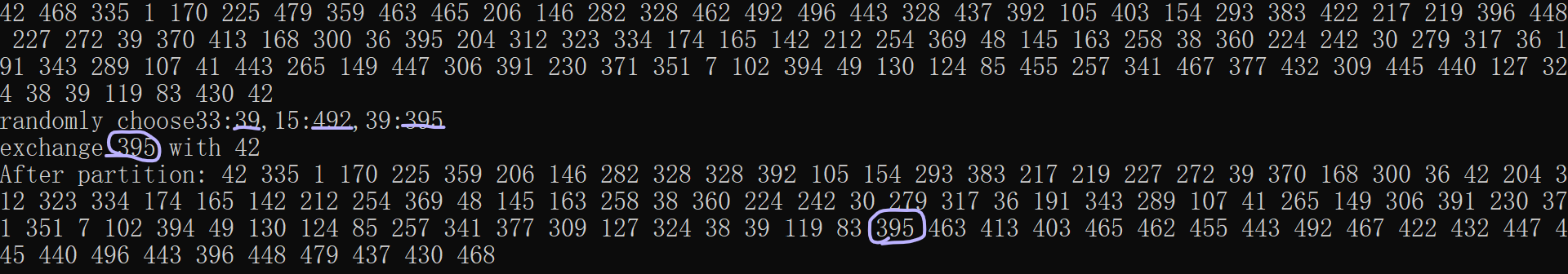
**In-class programming No.5**

Random quick sort

The quick sort algorithm performs better when the partitions are balanced. That is, key(A[r]) is closer to the middle after the partition.

One way to improve the algorithm is randomly choose 3 elements from the subarray, and take median as the key (swapping it with the rightmost element).

The algorithm can be illustrated as the following example:



***Requirements:***

* Generate an array of 100 random integers ranged from 1 to 500;
* Sort the array using the above algorithm;
* Print the sorting steps.
* Print your name in the end.

***Pseudocode:***

RANDOMIZED-PARTITION(A, p, r)

i,j,k ← RANDOM(p,r)

exchange A[r] ←→ MEDIAN(A[i], A[j], A[k])

return RANDOMIZED-PARTITION(A, p, r)

RANDOMIZED-QUICKSORT(A, p, r)

if p < r

then q ← RANDOMIZED-PARTITION(A, p, r)

RANDOMIZED-QUICKSORT(A, p, q-1)

RANDOMIZED-QUICKSORT(A, q+1, r)

Send the code and result to the TA’s email after class.