VE477 Lab7 Report

Student Name: Zhou Haoquan

Student ID: 519370910059

1. `Randomize Search`

Average number of indices that are picked, assuming the size of `A` is `n`

no index i such that A[i] = k	`O(nlogn)`
exactly one index i such that A[i] = k	`O(n)`
more than one index i such that A[i] = k	`O(n/m)`

Now we prove the results above:

1.1 no index i such that A[i] = k

Each time we choose an index, we have possibility of

$$P = \frac{n-i}{n}$$

to choose a new index, assuming `i` indices have already been chosen.

Then the expectation to choose that new index, knowing `i` indices have already been chosen, is

$$E = \frac{n}{n-i}$$

The expectation of total number of steps is then

$$\sum E = \sum_i \frac{n}{n-i} = n \sum \frac{1}{n-i} = \mathcal{O}(n \log n)$$

1.2 exactly one index i such that A[i] = k

The expectation can be derived from the equation

$$E(x) = 1 \times \frac{1}{n} + (1 - \frac{1}{n})(1 + E(x))$$

which results in

$$E(x) = n = \mathcal{O}(n)$$

1.3 more than one index i such that A[i] = k

The average case for this condition is just dividing the whole array into `O(m)` pieces. We can do this because the case is "average" case. Then we can write that the expectation is

$$E(x) = \mathcal{O}(\frac{n}{m})$$

2. `Linear Search`

Average case and worst case indices that are picked, assuming the size of `A` is `n`

Situation	Average Case	Worst Case
no index i such that A[i] = k	`O(n)`	`O(n)`
exactly one index i such that A[i] = k	`O(n)`	`O(n)`
more than one index i such that A[i] = k, assuming m indices	`O(n/m)`	`O(n - m)`

The proof is basically the same with the question one. And we omit it.

3. `Scramble Search`

Average case and worst case indices that are picked, assuming the size of `A` is `n`

Situation	Average Case	Worst Case
no index i such that A[i] = k	`O(n)`	`O(n)`
exactly one index i such that A[i] = k	`O(n)`	`O(n)`
more than one index i such that A[i] = k	`O(n/m)`	`O(n - m)`

The `scramble search` is actually equivalent to `linear search`.

Note that the `scramble search` has a smaller possibility to reach worst case than `linear search`.

4. Comparison

The `scramble search` is the best, since it has the best time complexity and the least possibility to reach the worst case.

5. Test Result

In the test, an random generated array with size `10000` is taken by three searching method and the result is shown as the following figure

```
Random search average time is:  2.439877978960673
Linear search average time is:  0.00044790903727213543
Scramble search average time is:  0.003887311617533366
```

we can see that the `linear search` has the best performance. The `Scramble search` is slower due to the shuffling process.