3) ~~Input | [...]~~ ,

a)

```
1  O = []
2  for i in 1 ... n do
3      index = Random(n - i + 1)
4      O[i] = I[index]              # I is input array
5      swap(index, n - i + 1)       # swap the chosen
                                    element with the last
                                    elements in the list
6  return O
```

b) Line 1: $O(1)$

Line 2: $O(n)$

Line 3-6: $O(1)$

We have a for-loop over all n numbers. In the loop, all operations take constant time, therefore the algorithm is in $O(n)$.

c) In each iteration, the algorithm picks an ~~random position / index~~ index of a number not yet chosen and inserts it at the i-th position in the output array. The probability for a number to be selected for the i-th position is $\frac{1}{n-i+1}$.

Therefore the probability for any output permutation is is:

$$\prod_{i=1}^{n} \frac{1}{n-i+1} = \prod_{i=0}^{n-1} \frac{1}{n-i} = \frac{1}{n} \cdot \frac{1}{n-1} \cdot \ldots \cdot \frac{1}{1} = \frac{1}{n!}$$

$\rightarrow$ uniform output prob. distribution

to avoid choosing duplicate elements.

~~This probability~~ At the end, we swap the currently selected element with the $n-i+1$-th element to ensure that all already selected elements are at the end of the input array