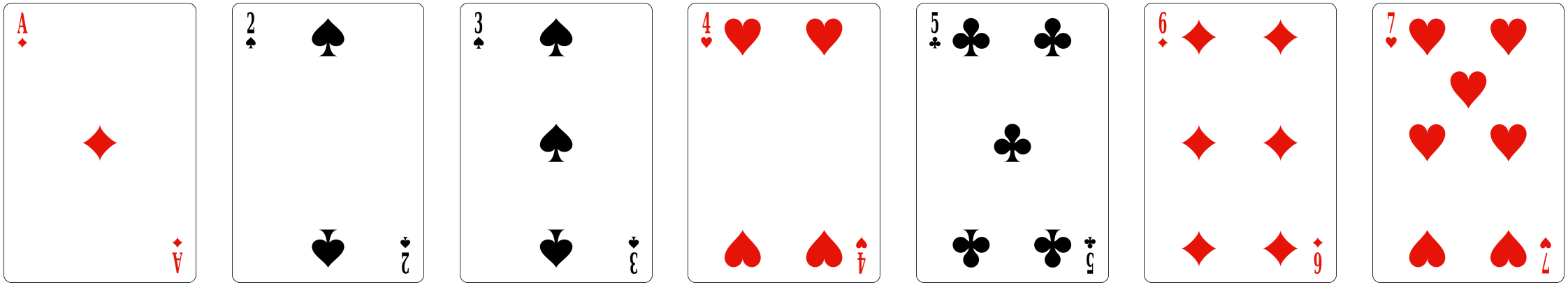
Random Permutation

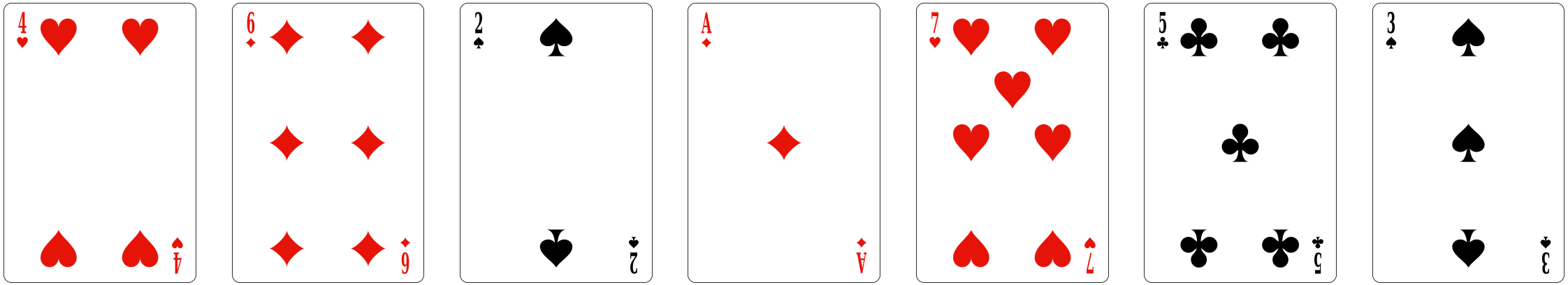
Shusen Wang

Random Permutation



Random Permutation

Now, the cards have random order.



What is uniform random permutation?

Number of Permutations

- The permutations of `{1, 2, 3}`:
  - `1, 2, 3.`
  - `1, 3, 2.`
  - `2, 1, 3.`
  - `2, 3, 1.`
  - `3, 1, 2.`
  - `3, 2, 1.`
- If a set contains n items, then there are $n!$ permutations.
- The factorial of n is

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1.$$

Number of Permutations

- The permutations of `{1, 2, 3}`:
  - `1, 2, 3.`
  - `1, 3, 2.`
  - `2, 1, 3.`
  - `2, 3, 1.`
  - `3, 1, 2.`
  - `3, 2, 1.`

There are $3! = 3 \times 2 \times 1 = 6$ possible arrangements.

- If the set contains n items, then there are $n!$ permutations.
- The factorial of n is

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1.$$

Uniform Random Permutations

- The permutations of `{1, 2, 3}`:
  - `1, 2, 3.`
  - `1, 3, 2.`
  - `2, 1, 3.`
  - `2, 3, 1.`
  - `3, 1, 2.`
  - `3, 2, 1.`

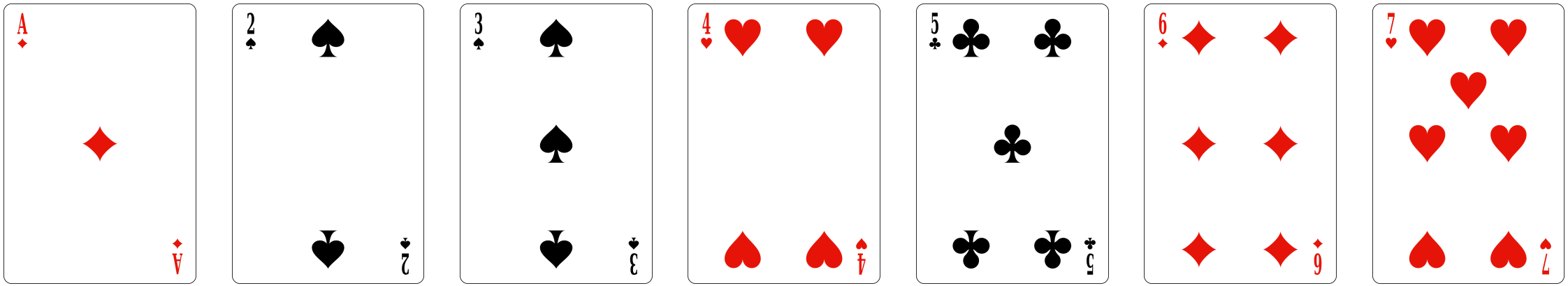
Sample one of $n!$ sequences uniformly at random.

Uniform random permutation means each of $n!$ sequences is equally likely.

Naively perform uniform random permutation

How to perform random permutation?

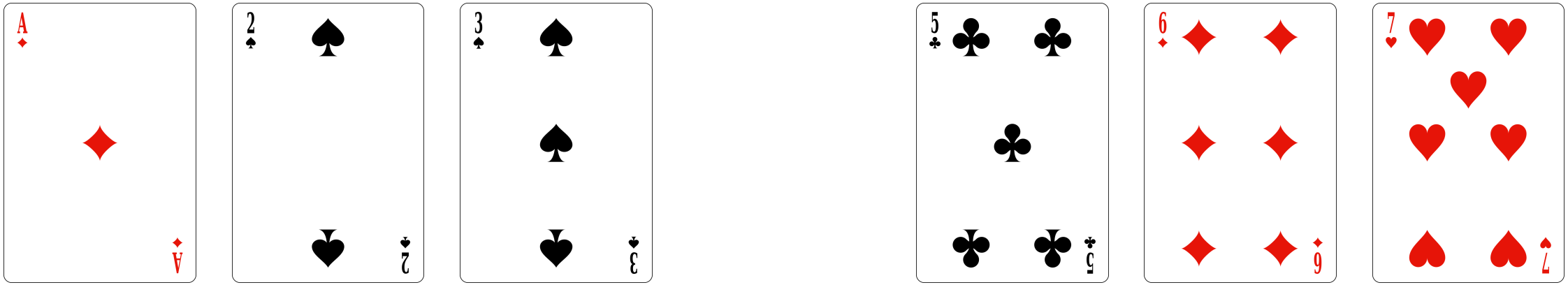
Sample a number from the set $\{1, 2, 3, 4, 5, 6, 7\}$ uniformly at random.

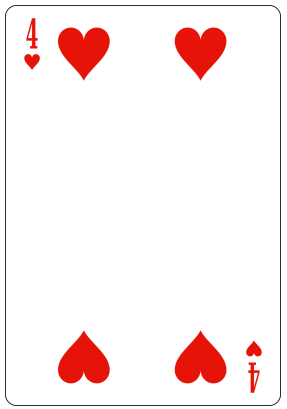


0th 1st 2nd 3rd 4th 5th 6th

How to perform random permutation?

Sample a number from the set $\{1, 2, 3, 5, 6, 7\}$ uniformly at random.

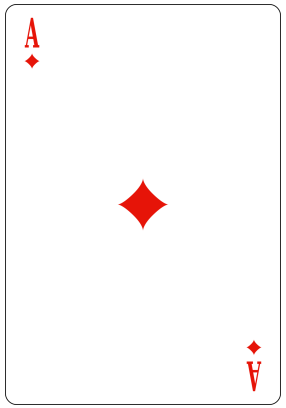
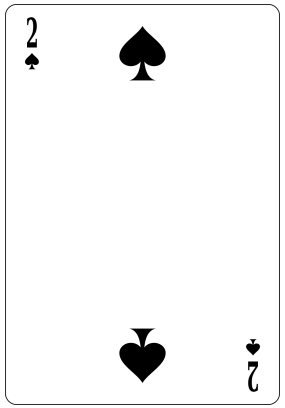
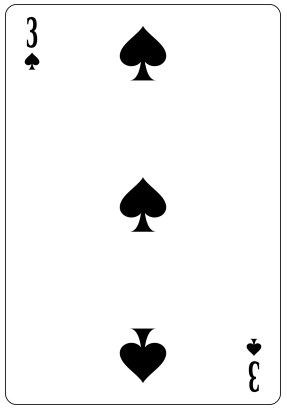
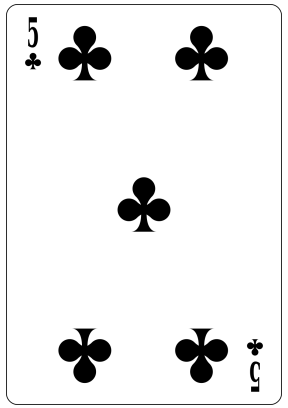
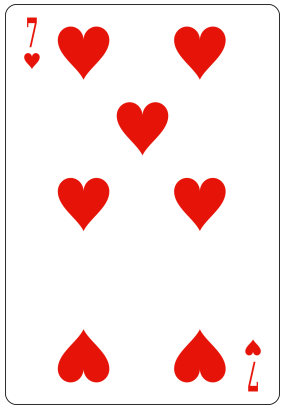
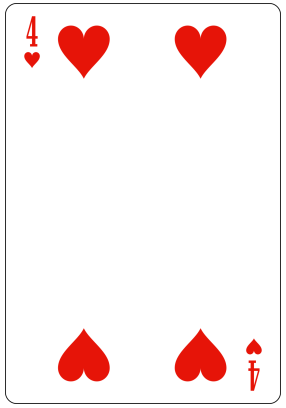
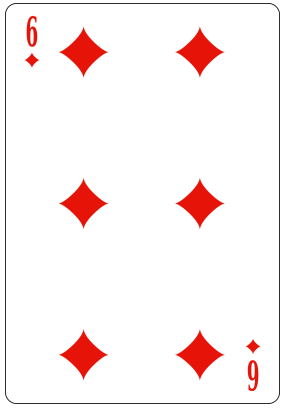




1st 2nd 3rd 4th 5th 6th

How to perform random permutation?

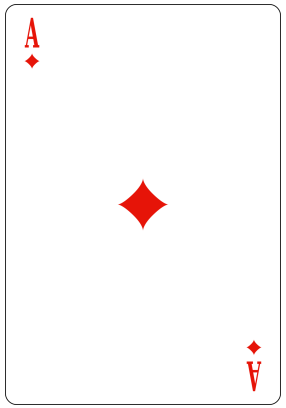
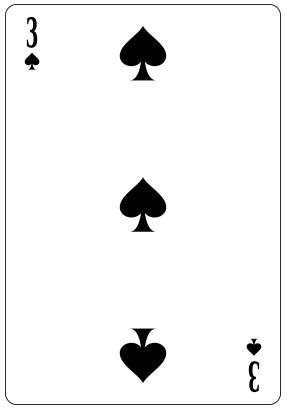
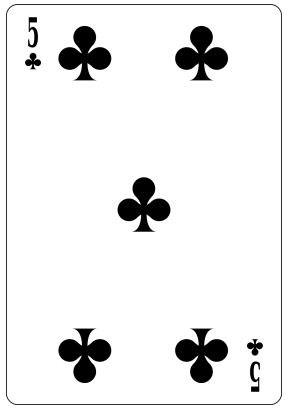
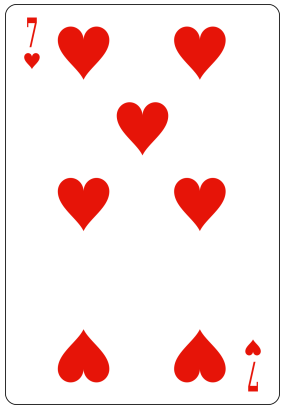
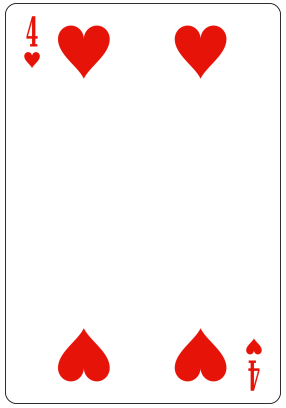
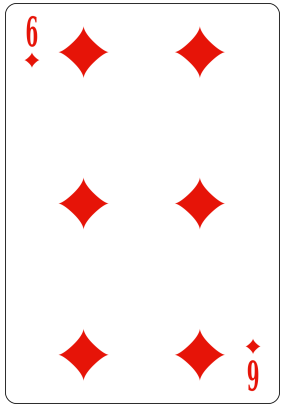
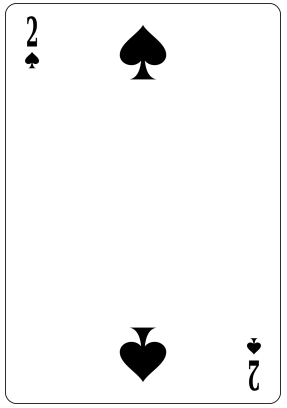
Sample a number from the set $\{1, 2, 3, 5, 7\}$ uniformly at random.

2nd 3rd 4th 5th 6th

How to perform random permutation?

Sample a number from the set $\{1, 3, 5, 7\}$ uniformly at random.

3rd

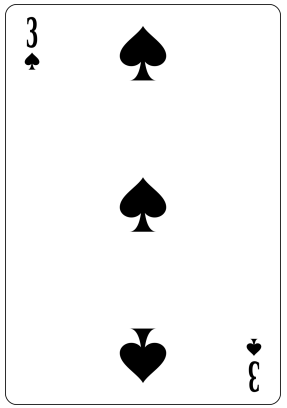
4th

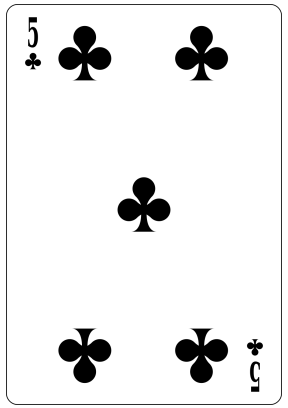
5th

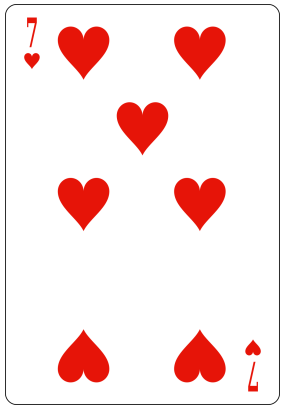
6th

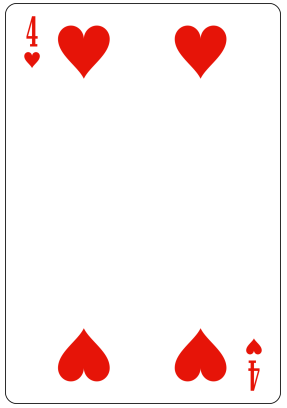
How to perform random permutation?

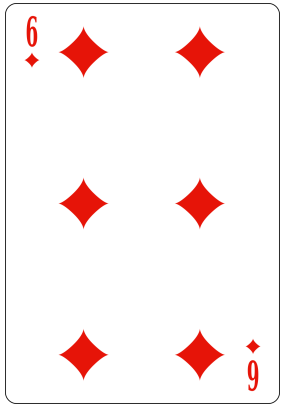
Sample a number from the set $\{3, 5, 7\}$ uniformly at random.

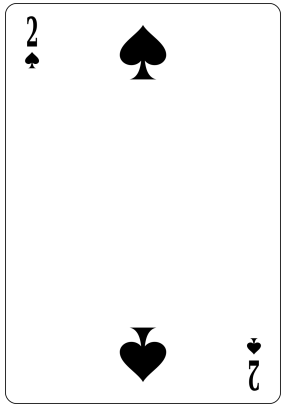


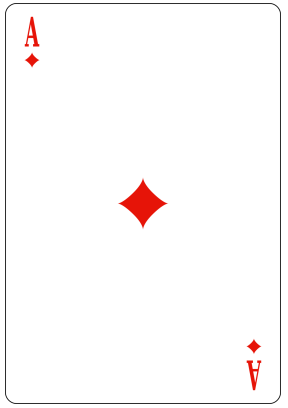












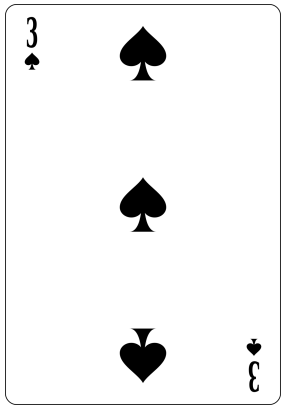
4th

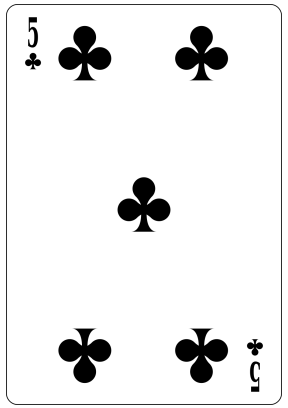
5th

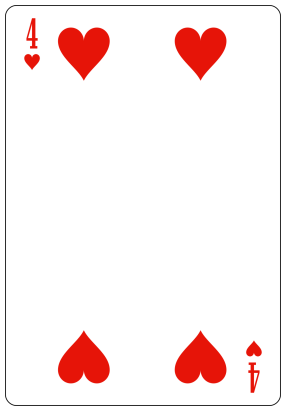
6th

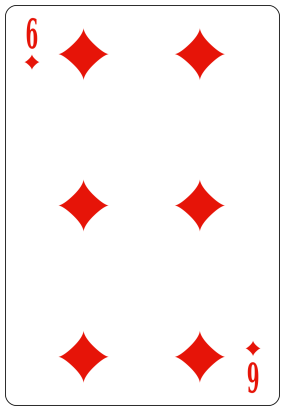
How to perform random permutation?

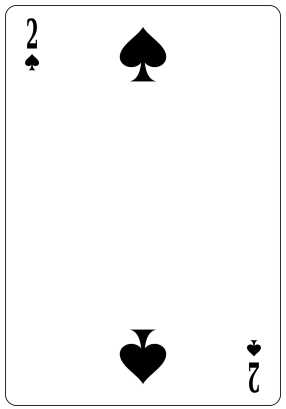
Sample a number from the set $\{3, 5\}$ uniformly at random.

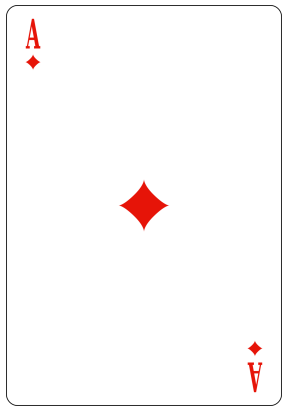


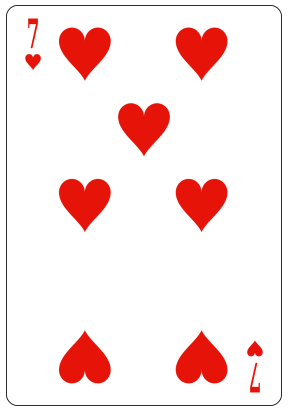










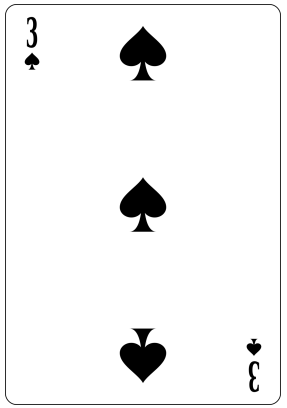


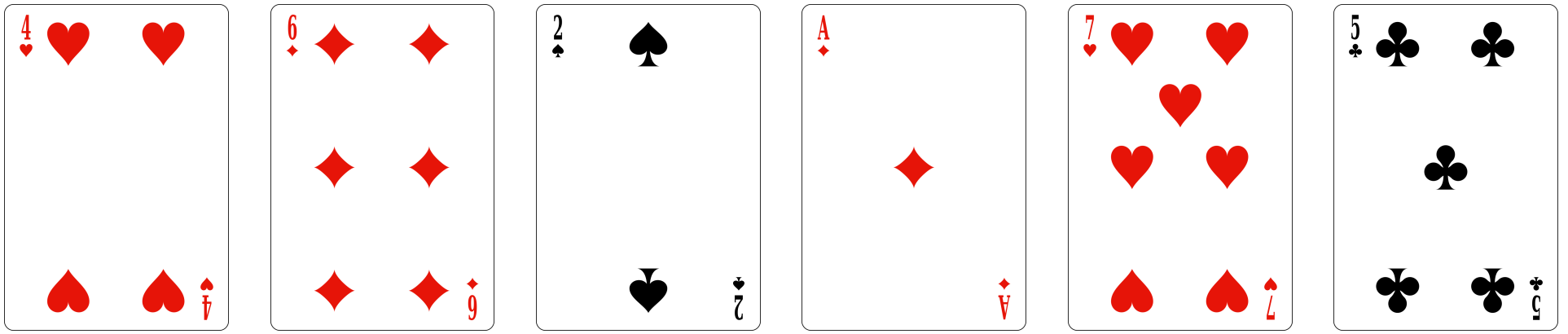
5th

6th

How to perform random permutation?

Put the remaining card at the end.

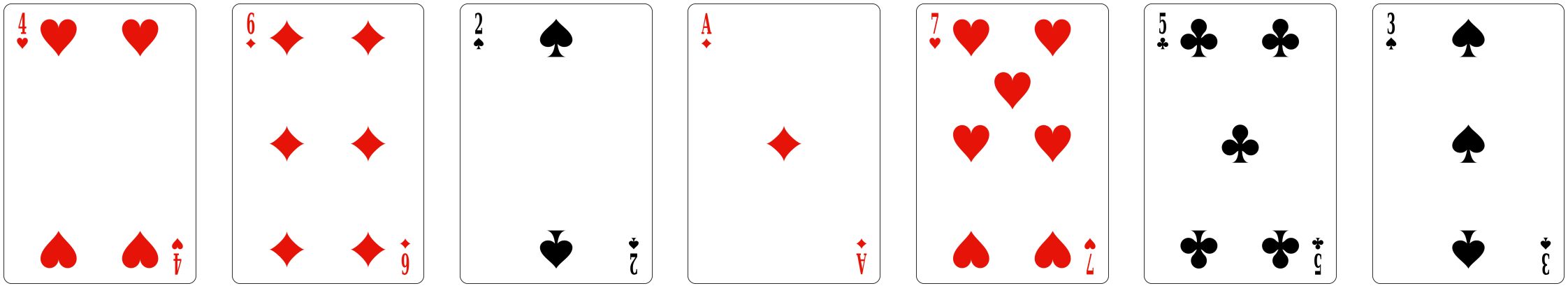




6th

How to perform random permutation?

Now, the sequence is a uniform random permutation.



Implementation

Initial State:

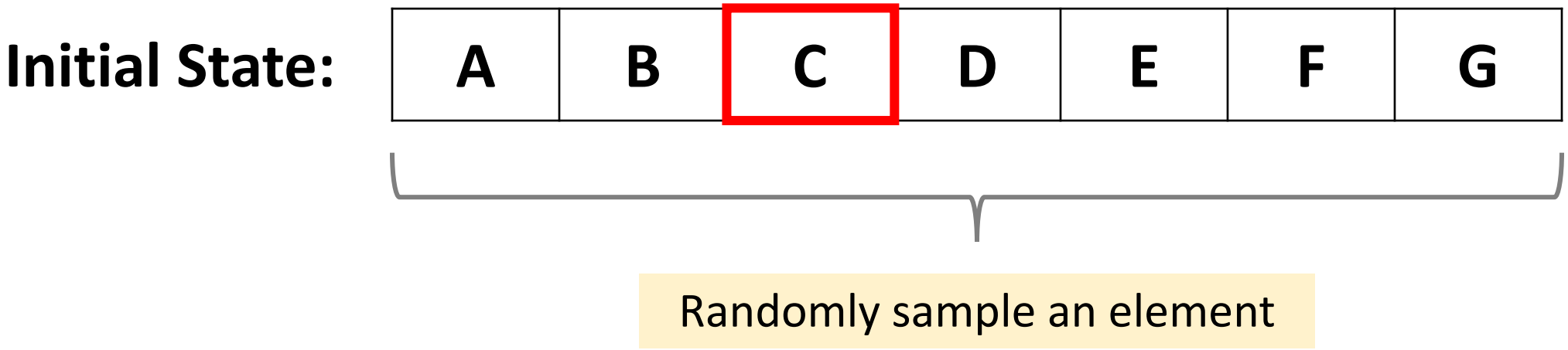
A	B	C	D	E	F	G

- Assume we have a random integer generator:

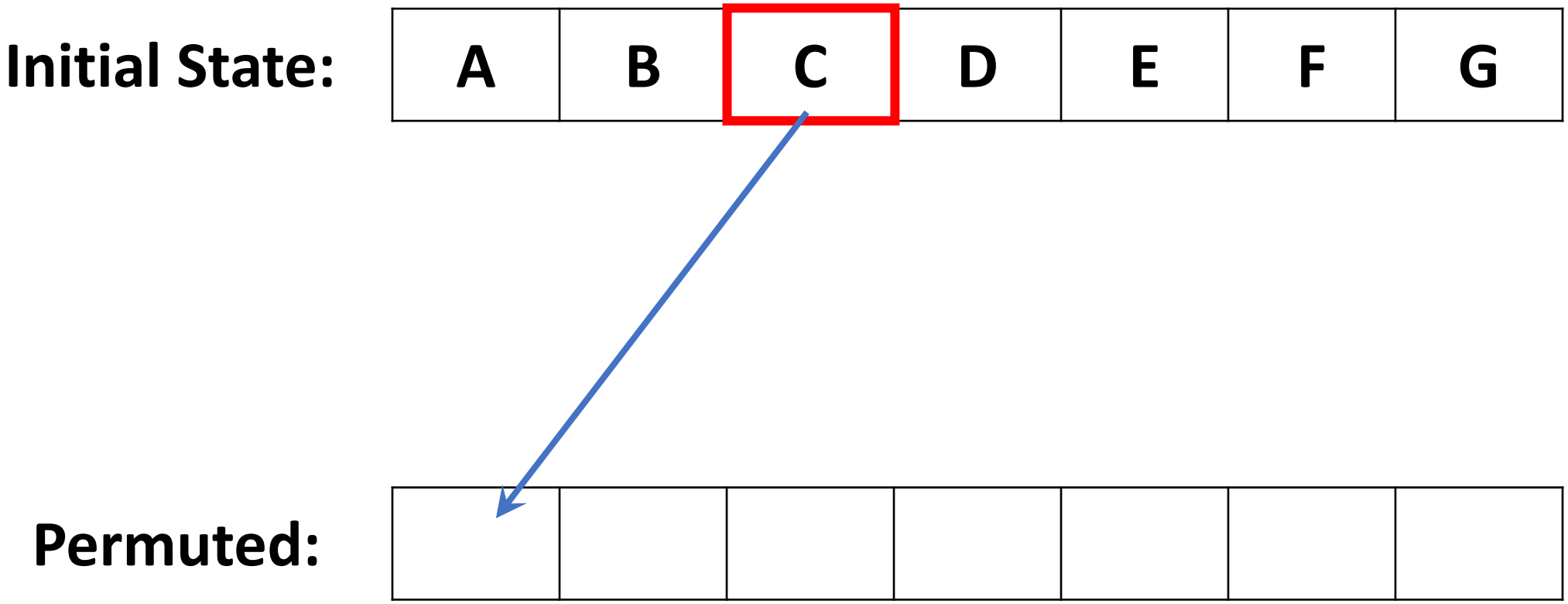
```
int k = uniform(int n);
```

- It samples one element from $\{0, 1, 2, \cdots, n-1\}$ uniformly at random.

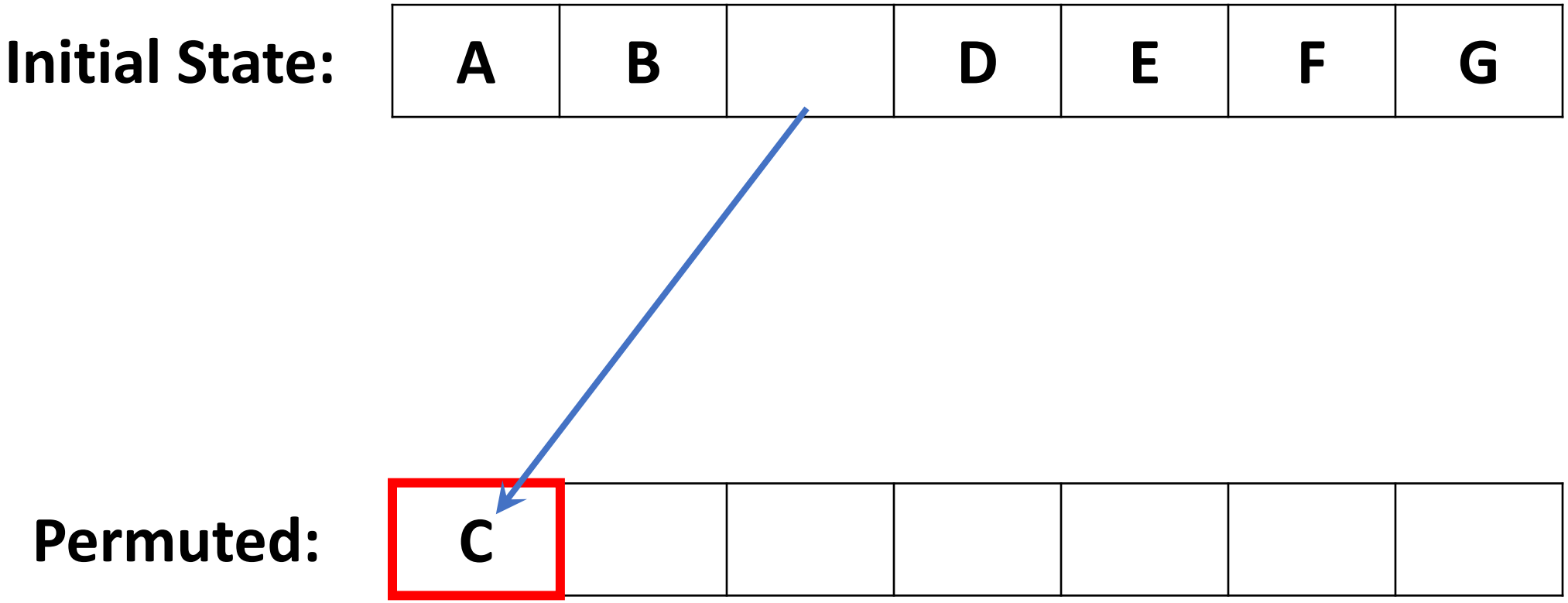
Implementation



Implementation

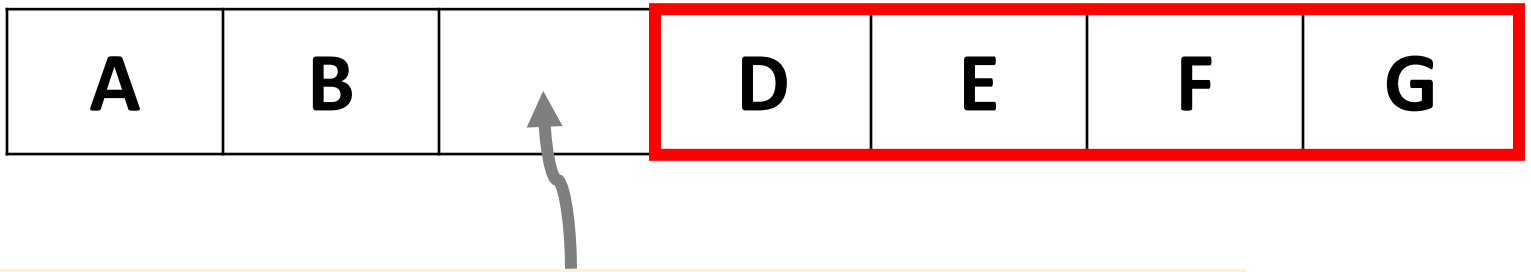


Implementation



Implementation

Initial State:

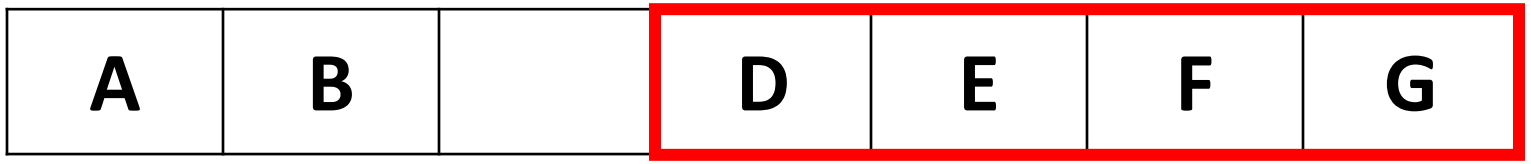


Fill the hole by moving the rest elements leftward.

Permuted:

Implementation

Initial State:



Permuted:

Implementation

After 1 Iteration:

A	B	D	E	F	G	

- $O(n)$ time complexity (on average) for sampling one element.
- Overall time complexity: $n + (n-1) + \cdots + 2 + 1 = O(n^2)$.

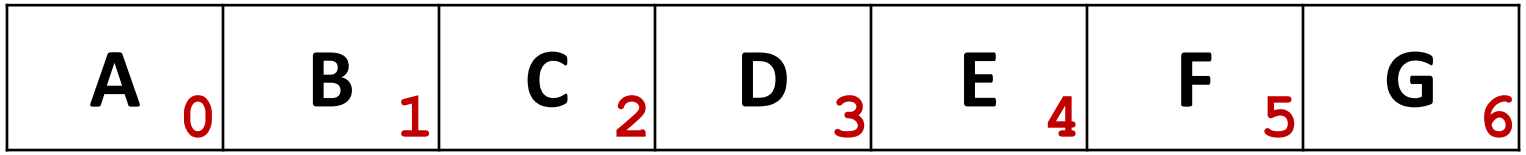
Permuted:

C						

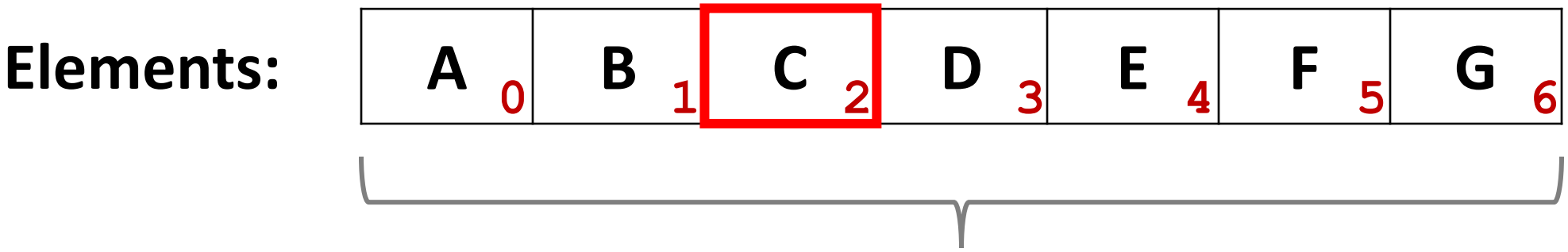
Fisher-Yates Shuffle

Initial State

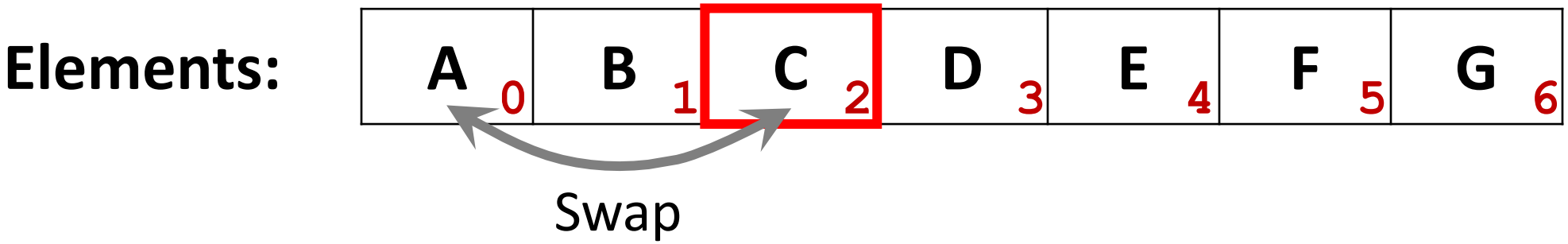
Elements:



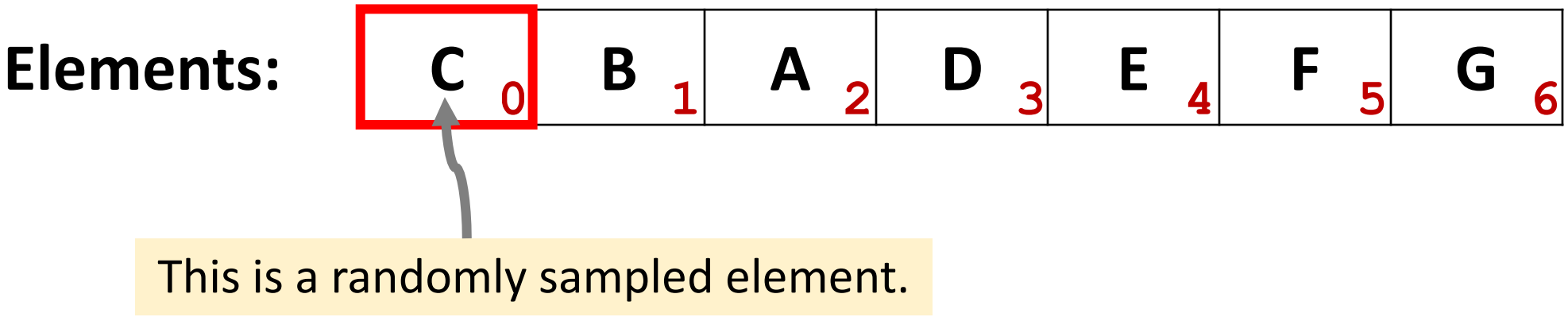
Iteration 0



Iteration 0

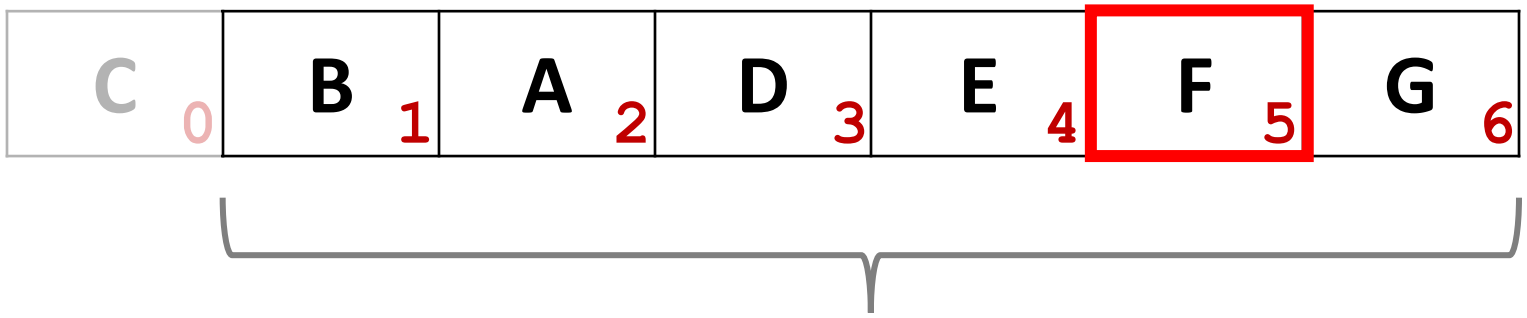


Iteration 0


Elements:
C 0
B 1
A 2
D 3
E 4
F 5
G 6
This is a randomly sampled element.

Iteration 1

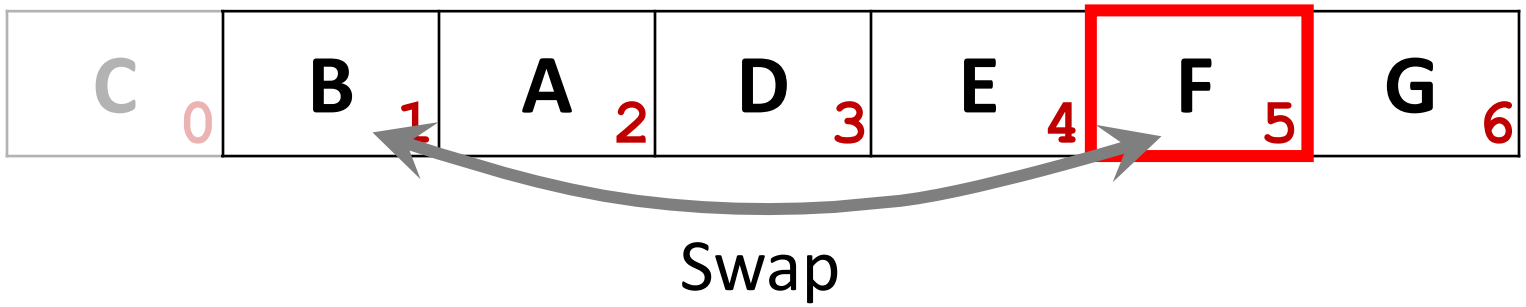
Elements:



Randomly sample an element

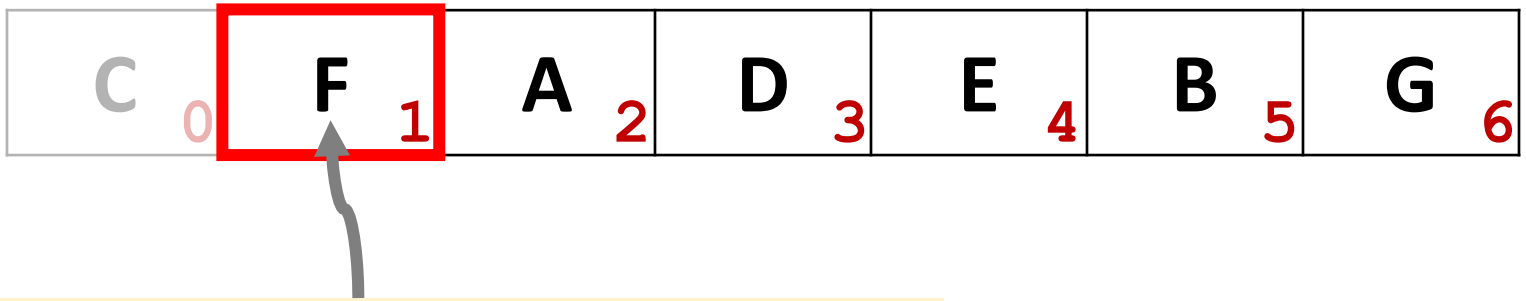
Iteration 1

Elements:



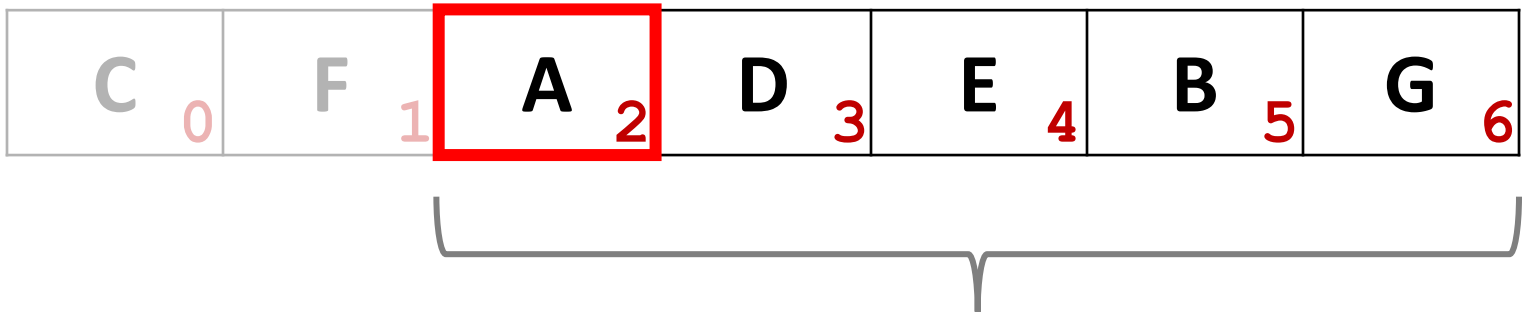
Iteration 1

Elements:



Iteration 2

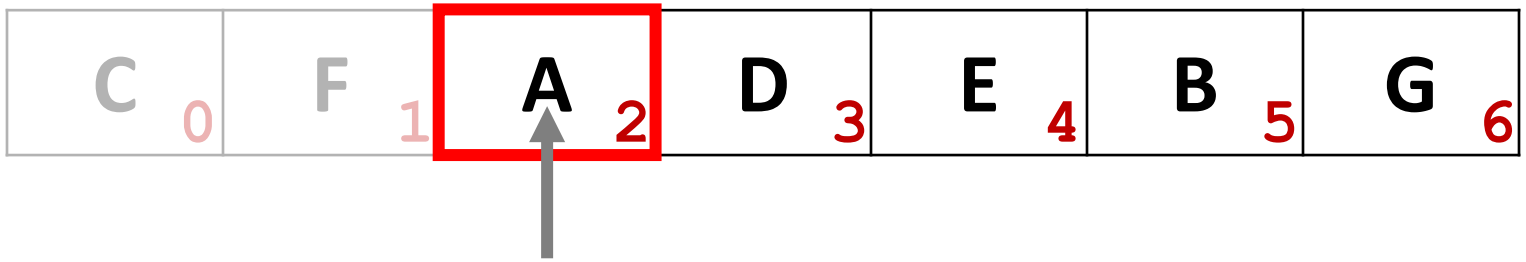
Elements:



Randomly sample an element

Iteration 2

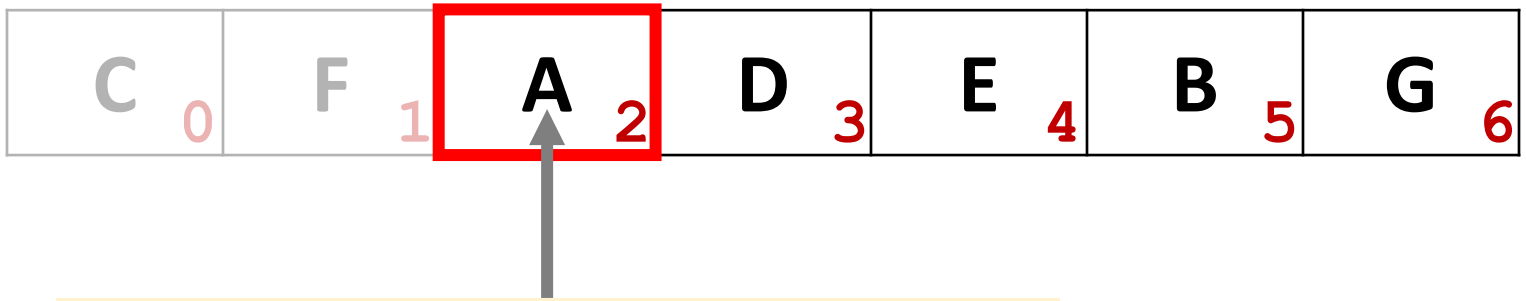
Elements:



Swap it with itself

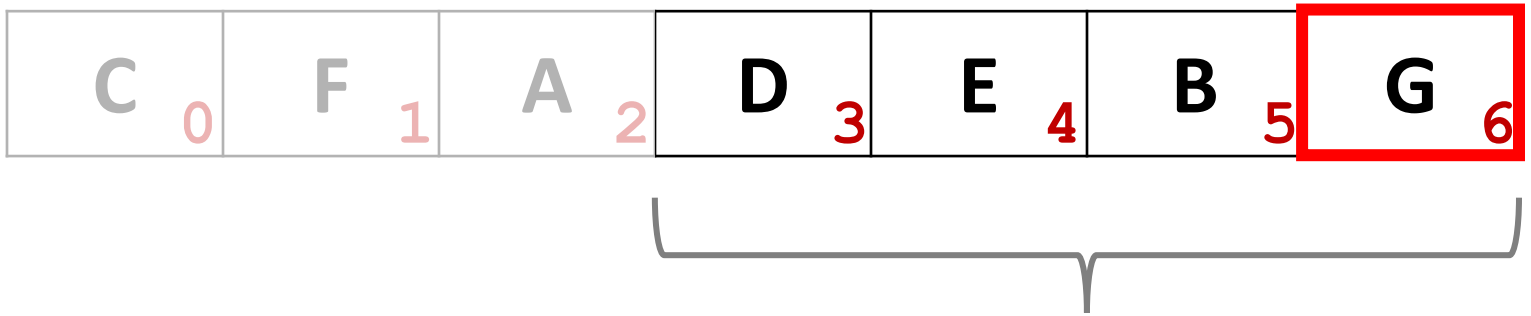
Iteration 2

Elements:



Iteration 3

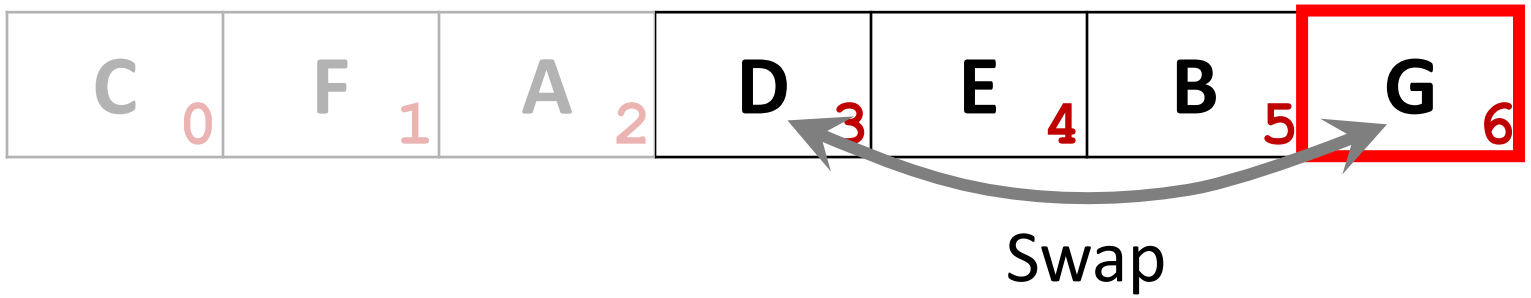
Elements:



Randomly sample an element

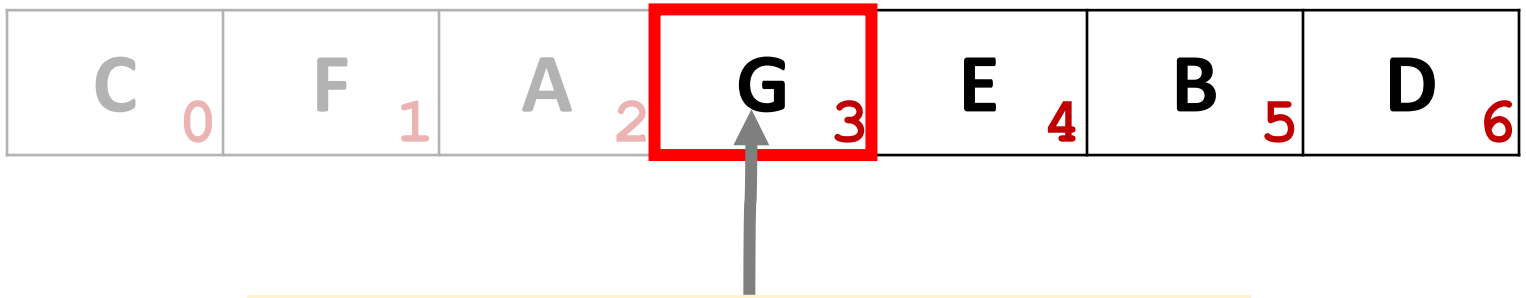
Iteration 3

Elements:



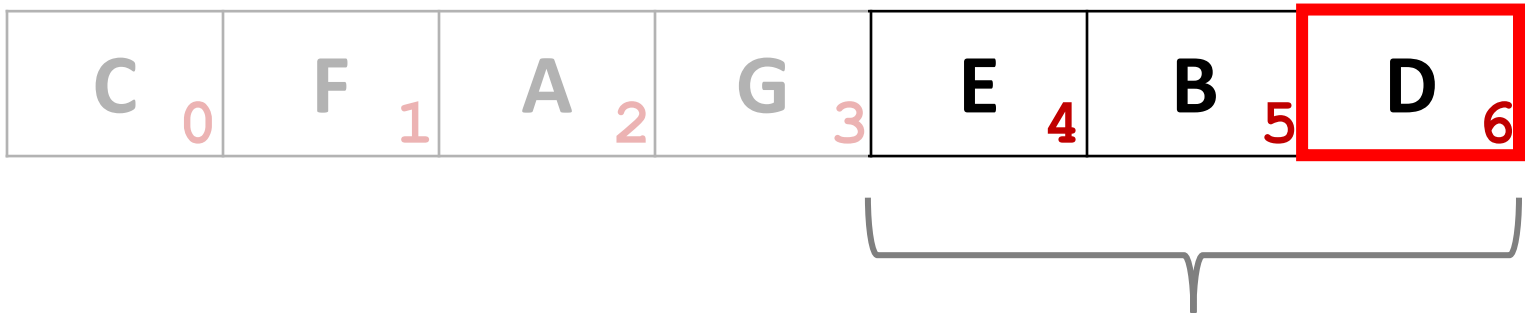
Iteration 3

Elements:



Iteration 4

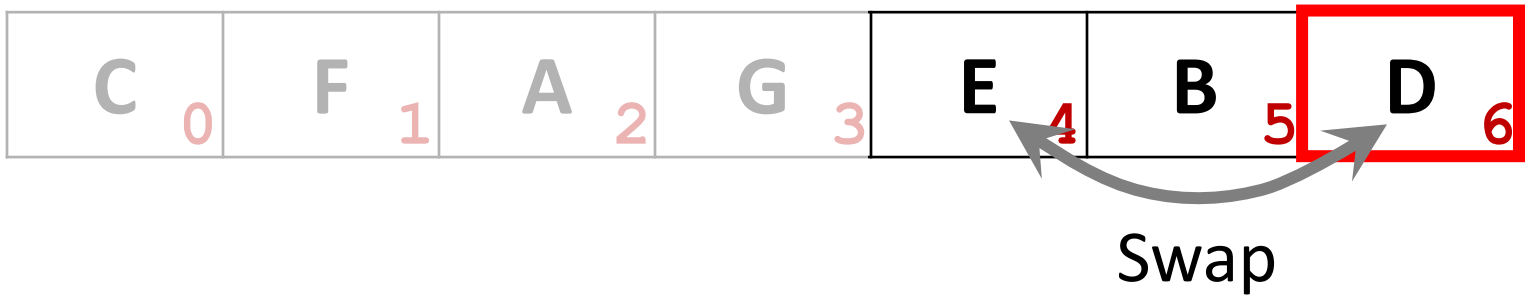
Elements:



Randomly sample an element

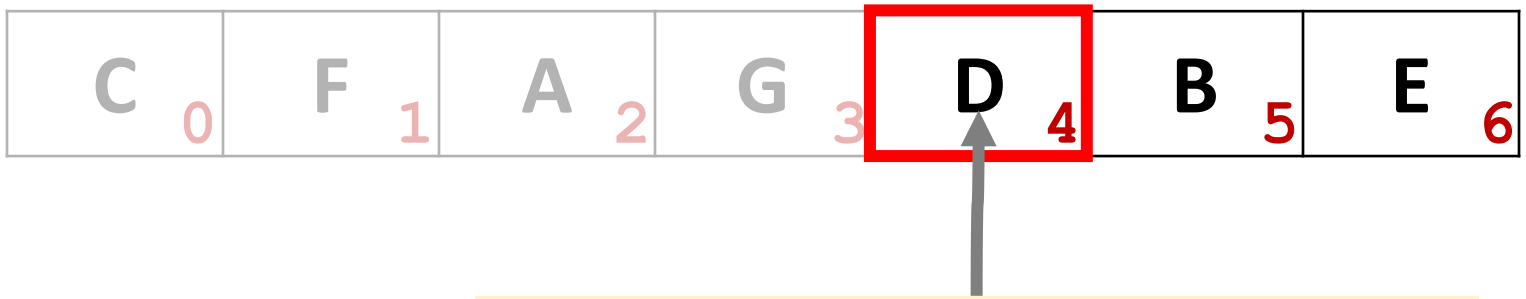
Iteration 4

Elements:



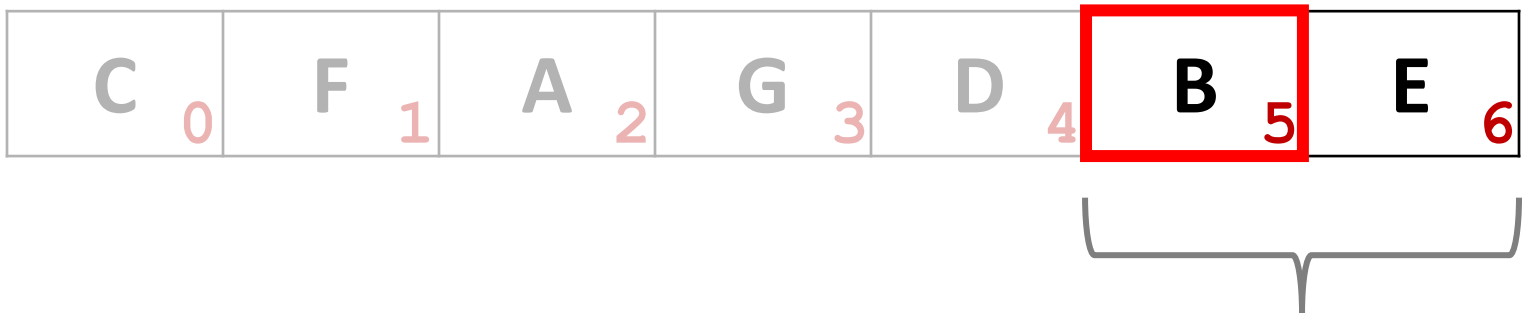
Iteration 4

Elements:



Iteration 5

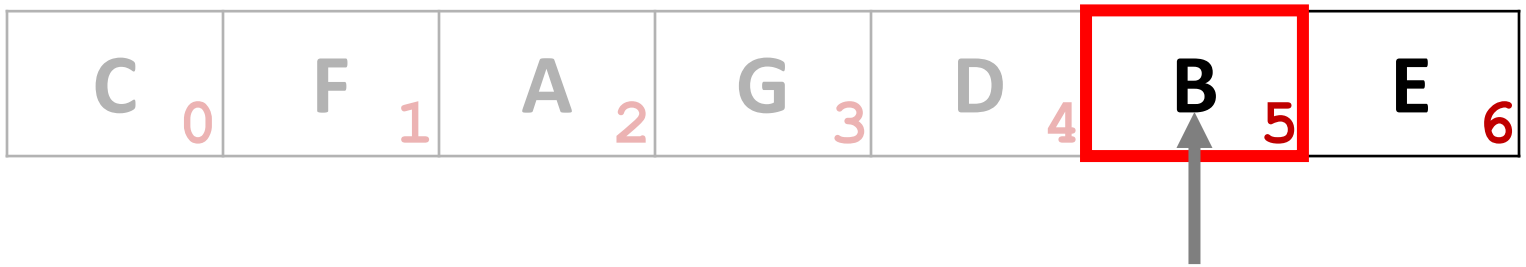
Elements:



Randomly sample an element

Iteration 5

Elements:



Iteration 5

Elements:

End of Procedure

Elements:

Leave the last element alone.

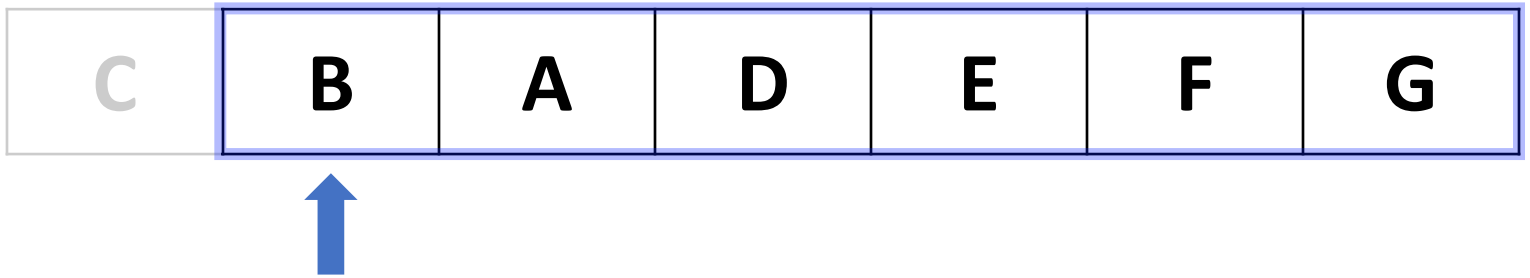
```
void permute(int arr[], int n) {
    int i;
    for (i=0; i <= n-2; i++) {
        // k is sampled from {0, 1, ..., n-i-1}
        int k = uniform(n-i);
        // j is in {i, i+1, ..., n-1}
        int j = i + k;
        // put arr[j] at the i-th position
        swap(arr, i, j);
    }
}
```

```
void permute(int arr[], int n) {
    int i;
    for (i=0; i <= n-2; i++) {
            // k is sampled from {0, 1, ..., n-i-1}
            int k = uniform(n-i);
            // j is in {i, i+1, ..., n-1}
            int j = i + k;
            // put arr[j] at the i-th position
            swap(arr, i, j);
    }
}
```

```
void permute(int arr[], int n) {
    int i;
    for (i=0; i <= n-2; i++) {
        // k is sampled from {0, 1, ..., n-i-1}
        int k = uniform(n-i);
        // j is in {i, i+1, ..., n-1}
        int j = i + k;
        // put arr[j] at the i-th position
        swap(arr, i, j);
    }
}
```

Explain the code

Elements:

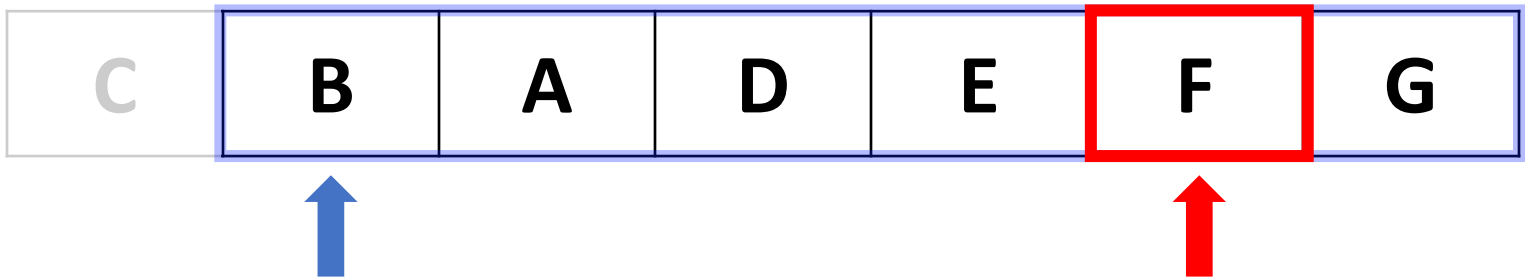


i is the current position

Currently, it is the i-th iteration.

Explain the code

Elements:

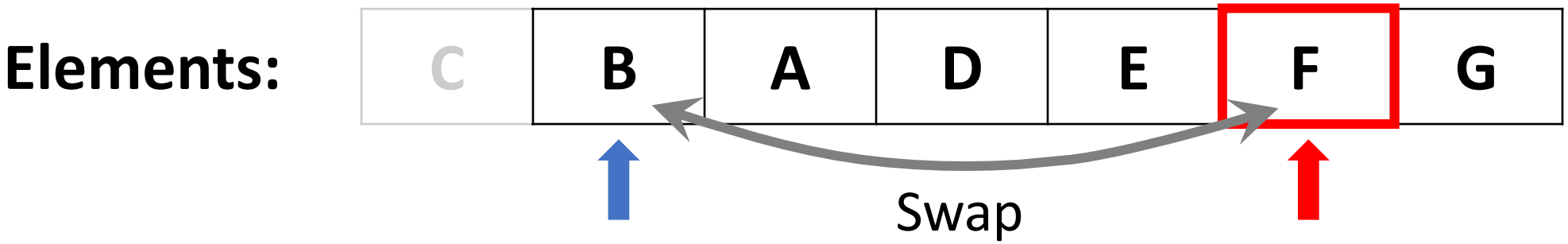


i is the current position

j is a random integer

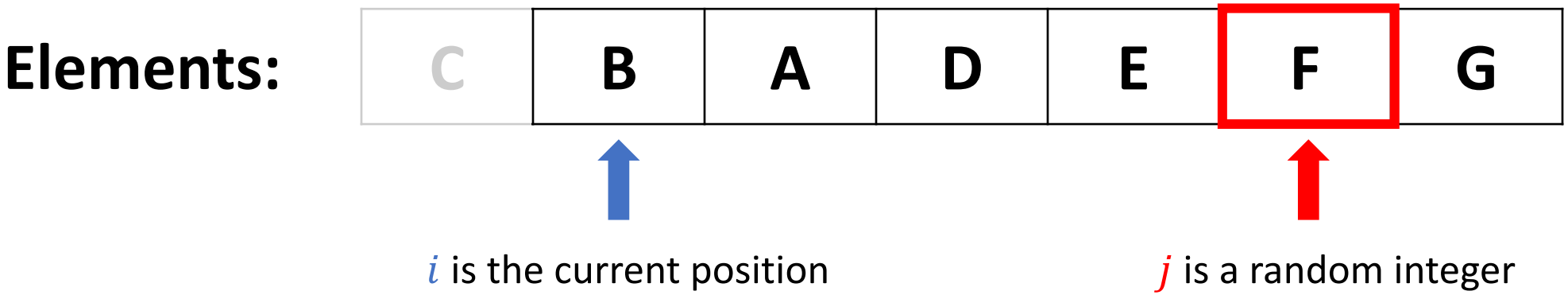
Currently, it is the i-th iteration.

Explain the code



Currently, it is the i-th iteration.

Time Complexity



- The per-iteration time complexity is $O(1)$.
- Totally $n - 1$ iterations.
- Thus, the overall time complexity is $O(n)$.

Thank You!