Selection Sort

Mr. Poole
Java

Selection Sort

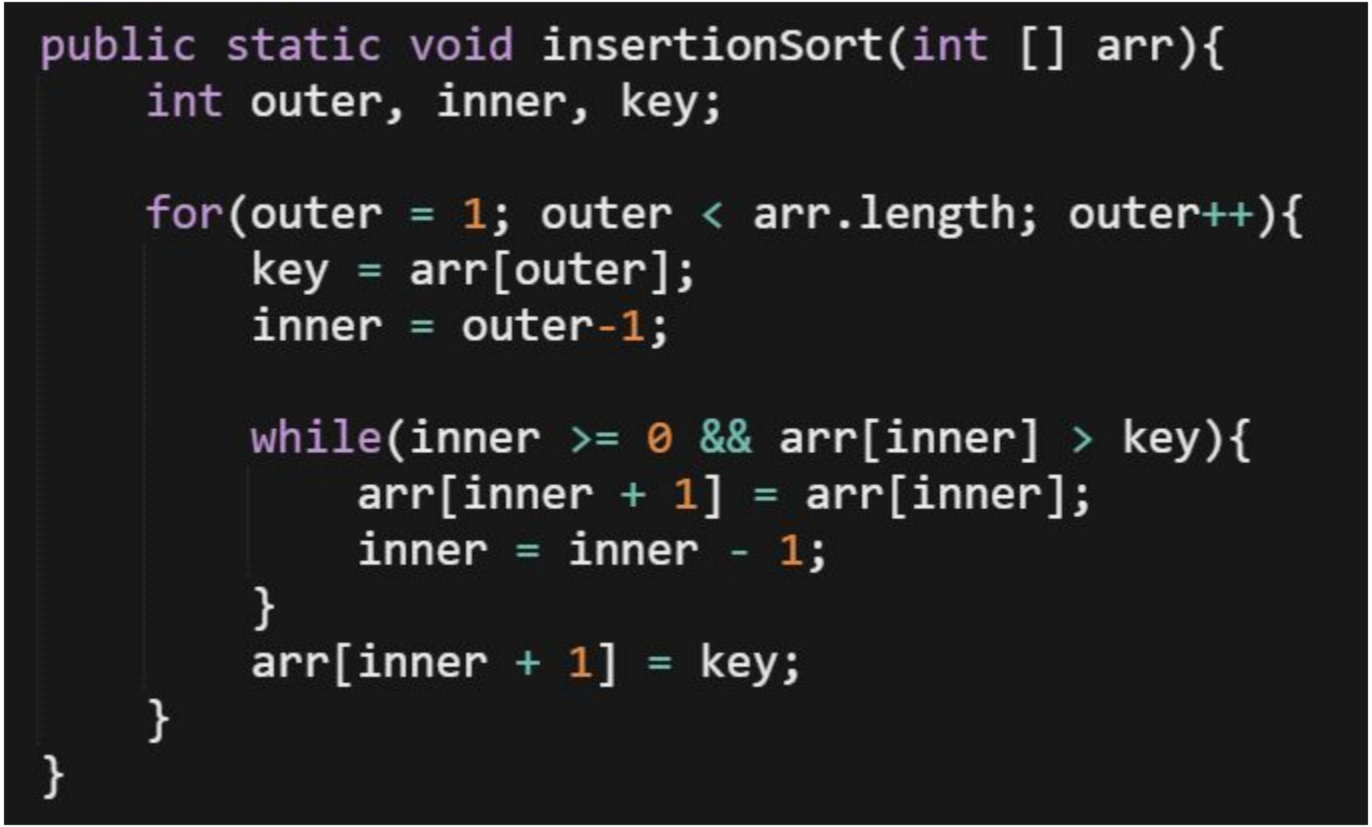
For each value in the array/ArrayList
Select the **SMALLEST** value and
move it to the **next front position**.

iteration	4	3	2	10	12	1	5	6

Example - Selection Sort **Swap Green Values**

iteration	4	3	2	10	12	1	5	6
0	**4**	3	2	10	12	**1**	5	6
1	1	**3**	**2**	10	12	4	5	6
2	1	2	**3**	10	12	4	5	6
3	1	2	3	**10**	12	**4**	5	6
4	1	2	3	4	**12**	10	**5**	6
5	1	2	3	4	5	**10**	12	**6**
6	1	2	3	4	5	6	**12**	**10**
7	1	2	3	4	5	6	10	**12**

Selection Sort - Code

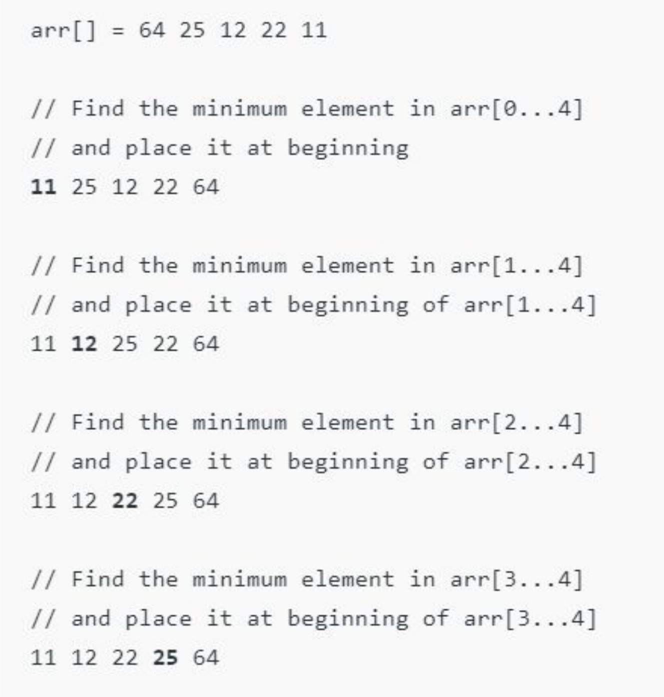


```
public static void insertionSort(int [] arr){
    int outer, inner, key;

    for(outer = 1; outer < arr.length; outer++){
        key = arr[outer];
        inner = outer-1;

        while(inner >= 0 && arr[inner] > key){
            arr[inner + 1] = arr[inner];
            inner = inner - 1;
        }
        arr[inner + 1] = key;
    }
}
```

Sorting - Selection Sort



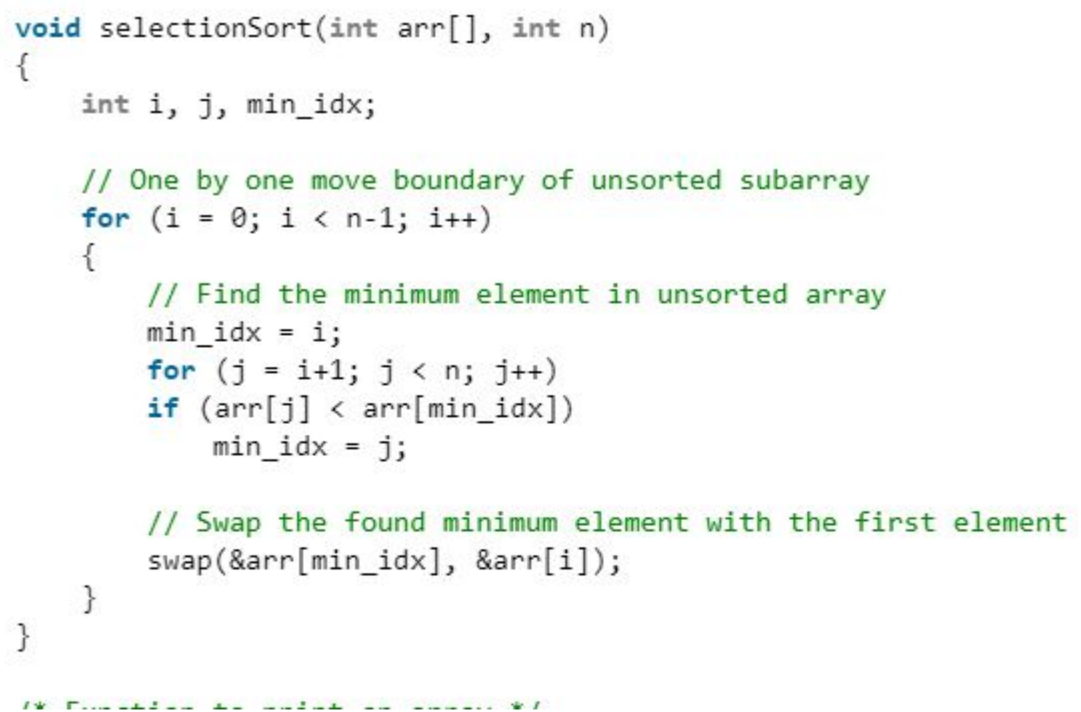
```
arr[] = 64 25 12 22 11

// Find the minimum element in arr[0...4]
// and place it at beginning
11 25 12 22 64

// Find the minimum element in arr[1...4]
// and place it at beginning of arr[1...4]
11 12 25 22 64

// Find the minimum element in arr[2...4]
// and place it at beginning of arr[2...4]
11 12 22 25 64

// Find the minimum element in arr[3...4]
// and place it at beginning of arr[3...4]
11 12 22 25 64
```



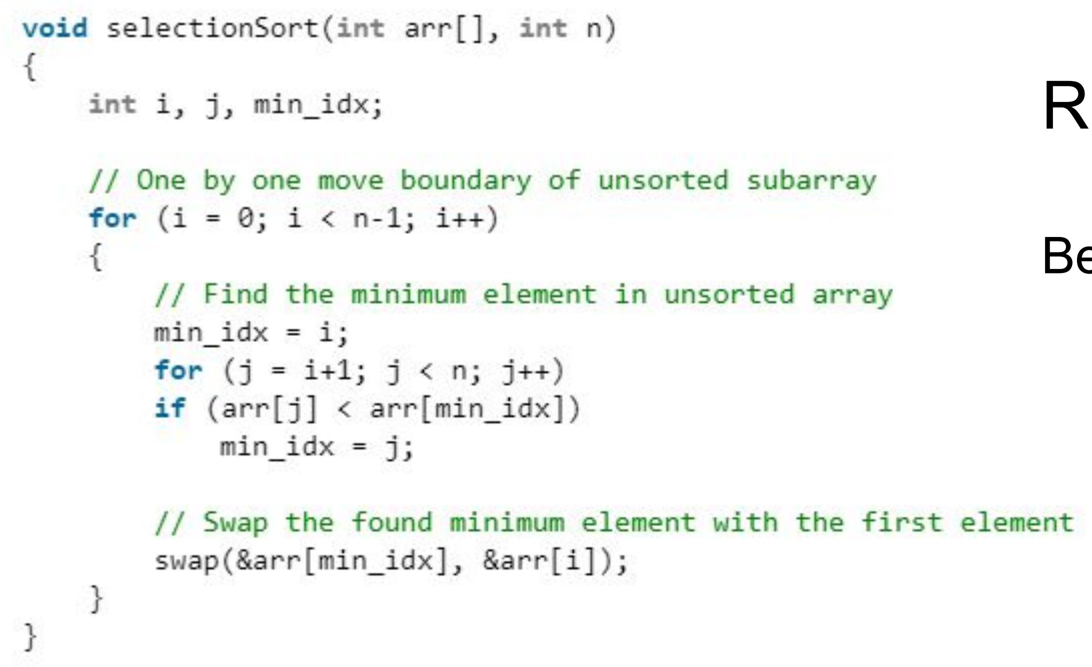
```
void selectionSort(int arr[], int n)
{
    int i, j, min_idx;

    // One by one move boundary of unsorted subarray
    for (i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted array
        min_idx = i;
        for (j = i+1; j < n; j++)
        if (arr[j] < arr[min_idx])
            min_idx = j;

        // Swap the found minimum element with the first element
        swap(&arr[min_idx], &arr[i]);
    }
}

/* Function to print an array */
```

Sorting - Selection Sort



```
void selectionSort(int arr[], int n)
{
    int i, j, min_idx;

    // One by one move boundary of unsorted subarray
    for (i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted array
        min_idx = i;
        for (j = i+1; j < n; j++)
        if (arr[j] < arr[min_idx])
            min_idx = j;

        // Swap the found minimum element with the first element
        swap(&arr[min_idx], &arr[i]);
    }
}
```

Runtime = $O(n^2)$

Because of 2 nested loops

Lab: Implement Selection Sort

Create an array of 200 random values,
Sort them with Selection Sort and print.