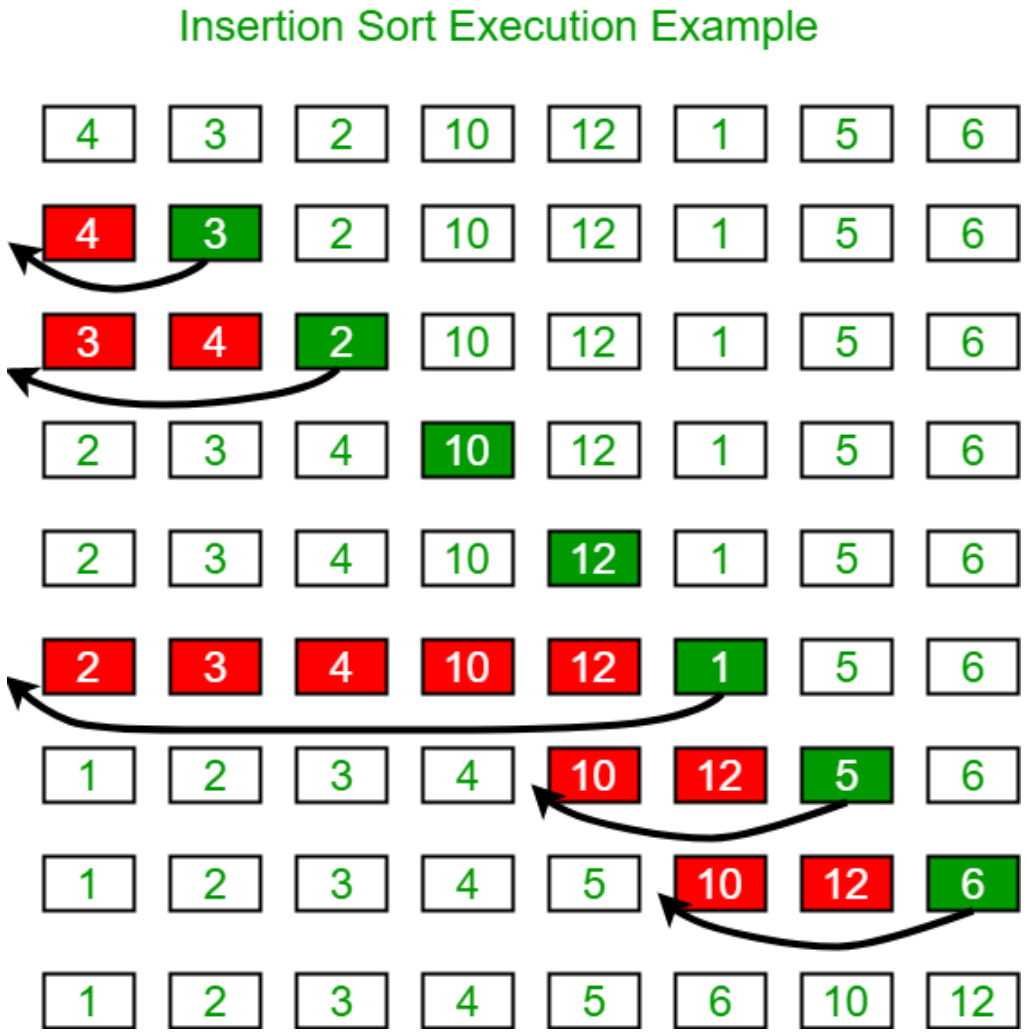
CS 111

arrays and functions

Insertion sort on an array

- Insertion sort is a sorting algorithm that works similarly to the way you sort playing cards in your hand
- The array is virtually split into a sorted part and an unsorted part
- Values from the unsorted part are picked and placed at the correct position in the sorted part

Example



https://www.geeksforgeeks.org/insertion-sort/

Insertion sort on an array

- Goal: Implement insertion sort on an array
- Method will be to loop *n* times
- On iteration *i* we make sure that the first *i* elements are sorted by inserting the i^{th} element into its correct place among earlier elements
- Note that we don't have to worry about the order of the earlier elements, because they were already sorted correctly after iteration *i* - 1

Plan

- The function is called **insertionSort**
- What parameters does it have?
- What return type does it have?
- What is the title line?

Plan

- For sorting, are there any smaller actions that we repeatedly need?
- Hint:

```
int temp = a;
a = b;
b = temp;
```

Plan

- The function is a loop with steps 0, 1, 2, . . . , $n - 1$
- The goal at *step* is to move element *step* back until it reaches its correct position
- The index of the element being moved back will gradually decrease as it moves back

Pseudocode

```
for (int step = 0; step < cap; step++){
    int index = step;
    int data = array[step];
    while(WE SHOULD MOVE ELEMENT BACK){
        SWAP index and index – 1 in the array;
        index--;
    }
}
```

Pieces of the puzzle

- What determines whether we should swap the data back?
  - data < array[index – 1]
- Could this test ever cause the program to crash?
  - Yes, when index = 0
- What is the condition for the while loop?
  - ((index > 0) && (data < array[index – 1]))