

## G5BADS: Bubble sort, selection sort and insertion sort

### 1. Bubble sort

```
void bubbleSort(int arr[]){
    int i;
    int j;
    int temp;

    for( i = arr.length-1; i > 0; i--){
        for(j = 0; j < i; j++){
            if( arr[j] > arr[j+1]){
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }// swap adjacent elements if in the wrong order
        }// end inner loop
    }//end outer loop
}// end bubble sort
```

### 2. Selection sort

```
void selectionSort(int arr[]){
    int i;
    int j;
    int temp;
    int pos_greatest;

    for( i = arr.length-1; i > 0; i--){
        pos_greatest = 0;
        for(j = 0; j <= i; j++){
            if( arr[j] > arr[pos_greatest]){
                pos_greatest = j;
            }
        }//end inner for loop
        temp = arr[i];
        arr[i] = arr[pos_greatest];
        arr[pos_greatest] = temp;
    }//end outer for loop
}//end selection sort
```

### 3. Insertion sort

```
void insertionSort(int arr[]){
    int i;
    int j;
    int temp;

    for(j=1; j < arr.length-1; j++){ // j is the leftmost unsorted index
        temp = arr[j]; // select leftmost unsorted element
        i = j; // i ranges over the sorted part
        while(i > 0 && arr[i-1] >= temp){ // until find a smaller element
            arr[i] = arr[i-1]; // shift elements to the right
            i--; // go left one position
        }
        arr[i] = temp; // insert selected element
    }
} // end insertion sort
```

Worst case time complexity of bubble sort:  
time usage function:  
complexity class:

Worst case time complexity of selection sort:  
time usage function:  
complexity class:

Worst case time complexity of insertion sort:  
time usage function:  
complexity class:

Please keep this handout for the lecture on loop invariants and algorithm correctness!