Bubble sort

Implementation

Bubble sort

```
for(int i = 0; i < n - 1; i++){
    for(int j = 0; j < n - 1 - i; j++){
        cout << i << " " << j << endl;
        if(arr[j] > arr[j + 1]){
            swap(arr[j], arr[j + 1]);
        }
    }
}
```

Bubble sort 2

```
    for(int i = 0; i < n; i++){
        for(int j = n - 1; j > i - 1; j--){
            cout << i << " " << j << endl;
            if(arr[j] < arr[j - 1]){
                swap(arr[j], arr[j + 1]);
            }
        }
    }
```

Bubble sort 3

```
for(int i = 1; i <= n - 1; i++){
        for(int j = 0; j <= n - i - 1; j++){
            cout << i << " " << j << endl;
            if(arr[j] > arr[j + 1]){
                swap(arr[j], arr[j + 1]);
            }
        }
    }
```

Selection sort

```
for(int i = 0; i < n - 1; i++){
    int small = i;
    for(int j = i + 1; j < n; j++){
        if(arr[j] < arr[small])
            small = j;
    }
    swap(arr[small], arr[i]);
}
```

Insertion Sort

```
for(int i = 1; i < n; i++){
    int cur = i, val = arr[i];
    while(cur > 0 && val < arr[--cur]){
        swap(arr[cur], arr[cur + 1]);
    }
}
```

Binary Search

```
int left = 0, right = n - 1, mid;
while(left <= right){
    mid = (left + right) / 2;
    if(arr[mid] > key){
        right = mid;
    }
    else if(arr[mid] < left){
        left = mid;
    }
    else{
        cout << "found at " << mid;
        break;
    }
}
```