Vježba Algoritmi

```
int binarySearch(int arr[], int l, int r, int x)
{
    while (l <= r) {
        int m = l + (r - l) / 2;

        if (arr[m] == x) {
            return m;
        }

        if (arr[m] < x) {
            l = m + 1;
        } else {
            r = m - 1;
        }
    }

    return -1;
}
```

```
int linearSearch(int arr[], int n, int x)
{
    for (int i = 0; i < n; i++) {
        if (arr[i] == x) {
            return i;
        }
    }
    return -1;
}
```

```
void matrixMultiply(int m1[r1][c1], int m2[r2][c2],
                int r1, int c1, int r2, int c2,
                int result[r1][c2])
{
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            result[i][j] = 0;

            for (int k = 0; k < r2; k++) {
                result[i][j] += m1[i][k] * m2[k][j];
            }
        }
    }
}
```

```
int[] intersectTwoArrays(int a1[], int a2[], int n, int m)
{
    int result[max(n, m)];
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (a1[i] == a2[j]) {
                result[k++] = a[i];
            }
        }
    }
    return result;
}
```

```
int[] mergeTwoSortedArrays(int a1[], int a2[], int n, int m)
{
    int result[n+m];
    int i = 0, j = 0;
    int k = 0;

    while (i < n && j < m) {
        if (a1[i] < a2[j]) {
            result[k++] = a1[i++];
        } else {
            result[k++] = a2[j++];
        }
    }

    while (i < n) {
        result[k++] = a1[i++];
    }

    while (j < m) {
        result[k++] = a2[i++];
    }

    return result;
}
```

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

    for (i = 0; i < n - 1; i++) {

        min_idx = i;
        for (j = i + 1; j < n; j++) {
            if (arr[j] < arr[min_idx])
                min_idx = j;
        }

        if (min_idx != i) {
            int tmp = arr[min_idx];
            arr[min_idx] = arr[i];
            arr[i] = tmp;
        }
    }
}
```

```
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;

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

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    bool swapped;
    for (i = 0; i < n - 1; i++) {
        swapped = false;
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                swap(arr[j], arr[j + 1]);
                swapped = true;
            }
        }

        if (swapped == false)
            break;
    }
}
```