Programming 1

Tutorial 12

Activity 1

Implement the selection sort algorithm from the given pseudocode below:

Function: Selection sort
Input: A non-empty integer array A
Output: An integer array B which is the sorted version of array A in ascending order

```
1.  Let B be an array, |B| = |A|
2.  For i = 0 to |A| - 1
3.      p ← i
4.      For j = i + 1 to |A| - 1
5.          If A[j] < A[p] Then
6.              p ← j
7.          End If
8.      End For
9.      B[i] ← A[p]
10.     A[p] ← A[i]
11. End For
12. Return B
```

Create a class named `SelectionSortDemo`. Implement this algorithm in a static method named `selectionSort` and write some code in the `main` method to test the algorithm.

Activity 2

Implement the binary search algorithm from the given pseudocode below:

Function: binarySearch(A, k, low, high)
Input: Integer array A
Integer k
Integer low
Integer high
Output: Position of k in A or -1 if A does not contain A

```
1.  If high < low Then
2.      Return -1
3.  End If
4.  mid ← (low + high) ÷ 2
5.  If A[mid] = k Then
6.      Return mid
7.  Else If A[mid] > k Then
8.      Return binarySearch(A, k, low, mid – 1)
9.  Else
10.     Return binarySearch(A, k, mid + 1, high)
11. End If
```

Create a class named `BinarySearchDemo`. Implement this algorithm in a static method named `binarySearch` and write some code in the `main` method to test the algorithm.

Activity 3

Implement the algorithm to find the intersection of two integer arrays from pseudocode:

Function: Intersect
Input: Two finite sets A, B
Output: A finite set C such that C = A ∩ B

1. C ← Ø
2. **If** |A| > |B|
3. **Then** Swap(A, B)
4. **End**
5. **For every** x ∈ A **Do**
6. **If** x ∈ B
7. **Then** C ← C ∪ {x}
8. **End**
9. **End**
10. **Return** C

`A`, `B` and `C` are sets, so the suitable data structure for them is `Set` (`HashSet`, `TreeSet`, your choice). Create a class named `ArrayIntersectionDemo`. Implement this algorithm in a static method named `intersect` and write some code in the `main` method to test the algorithm.