

Exam – extra part: solution

Tasks: solutions

Task 1 (4 points + 3 points)

a) (4 points)

```
public static void sort (int[] numbers)
{
    int    lastPos = numbers.length - 1;
    for (int pos = 0; pos < lastPos; pos++)
    {
        for (int p = lastPos; p > pos; p--)
        {
            if (numbers[p] < numbers[p - 1])
            {
                int    t = numbers[p - 1];
                numbers[p - 1] = numbers[p];
                numbers[p] = t;
            }
        }
    }
}
```

b) (3 points)

In the first pass through the main loop there are $n - 1$ comparisons, in the second pass there are $n - 2$ comparisons, and so on. In the last pass there is only 1 comparison. The total number of comparisons are:

$$(n - 1) + (n - 2) + \dots + 2 + 1 = n(n - 1) / 2$$

This means that the time complexity of the algorithm in terms of the number of comparisons, can be given by the following complexity function:

$$t(n) = n(n - 1) / 2$$

Task 2 (1 point + 1 point + 1 point)

a) (1 point)

$$u(n) \in \theta(n)$$

b) (1 point)

$$v(n) \in \theta(n \log_2 n).$$

c) (1 point)

$$w(n) \in \theta(n^2)$$

Task 3 (4 points + 3 points + 3 points)

a) (4 points)

```
private class ListIterator implements Iterator
```

```
{
    private Node    current = null;

    public ListIterator ()
    {
        current = first;
    }

    public boolean hasElement ()
    {
        return current != null;
    }

    public int element () throws java.util.NoSuchElementException
    {
        if (!this.hasElement ())
            throw new java.util.NoSuchElementException (
                "end of the iteration");

        int    element = current.value;

        return element;
    }

    public void move ()
    {
        current = current.next;
    }
}
```

b) (3 points)

```
// add adds a given integer to the end of the list.
public void add (int value)
{
    Node    n = new Node (value);
    if (first == null)
    {
        first = n;
        last = n;
    }
    else
    {
        last.next = n;
        last = n;
    }
}
```

c) (3 points)

```
List    list = new List ();
for (int i = 1; i <= 5; i++)
    list.add (i);

Iterator    iterator = list.iterator ();
while (iterator.hasElement ())
{
    int    element = iterator.element ();
    System.out.print (element + " ");

    iterator.move ();
}
System.out.println ();
```