# HW #2. Sort (Binary Input)

(Due: 3/15/2015)

The problem to be solved in this homework is exactly the same as HW1. You are given a set of integer sequences, and your program has to sort each of the sequences in ascending order. The only difference here is that HW2 uses binary formats for `input.txt` and `output.txt`. Specifically, the data in `input.txt` and `output.txt` are serialized in the MessagePack[1] binary format. You may notice that it is actually easier to handle the input/output data with the MessagePack deserialization / serialization APIs. You may also notice that processing of the binary data is more efficient than processing of the text data as in HW1.

## input.txt

{N:an **integer-type object** indicating the number of sequences in the input file},

{$SEQ_{INPUT\_1}$ : an **array-type object** corresponding to the first sequence of integers to be sorted},

{$SEQ_{INPUT\_2}$: an **array-type object** corresponding to the second sequence of integers to be sorted },

…

{$SEQ_{INPUT\_N}$: an **array-type object** corresponding to the $N^{th}$ sequence of integers to be sorted }

Each object is stored in binary. The objects are placed back-to-back in the file. Please refer to [2] for the object types in MessagePack

Following is the corresponding deserialization code in Java

```java
MessageUnpacker unpacker =
        MessagePack.newDefaultUnpacker(
                new FileInputStream("input.txt"));

int k;
int N = unpacker.unpackInt();


for ( k = 0; k < N; k++) {
    int len = unpacker.unpackArrayHeader();

    int[] numbers = new int[len];

    for ( m = 0; m < len; m++)
        numbers[m] = unpacker.unpackInt();

}

packer.close();
```

## output.txt

{SEQ$_{OUTPUT\_1}$: an array object containing the integer numbers of SEQ$_{INPUT\_1}$ sorted in ascending order},

{SEQ$_{OUTPUT\_2}$: an array object containing the integer numbers of SEQ$_{INPUT\_2}$ sorted in ascending order},

…

{SEQ$_{OUTPUT\_N}$: an array object containing the integer numbers of SEQ$_{INPUT\_N}$ sorted in ascending order},

Following is the corresponding serialization code in Java

```java
MessagePacker packer = MessagePack.newDefaultPacker(
                    new FileOutputStream("output.txt"));

for ( k = 0; k < N; k++) {
    ....
    int[] numbers = new int[len];
    ....

    packer.packArrayHeader(len);

    for ( m = 0; m<len; m++)
        packer.packInt(numbers[m]);
}

packer.close();
```

## Reference

[1]     Msgpack.org. *MesssagePack*. Available: http://msgpack.org/
[2]     *MessagePack object types*. Available:
        https://github.com/msgpack/msgpack/blob/master/spec.md#types