HW7 Requirement

Keyword

- A keyword is a tuple of [String name, Integer count, Double weight]
- For example:

  {

  name: “Fang”,

  count: 3,

  Weight: 5.5

  }

- A key word should output in format **[name,count,weight]** :

```
[Fang,3,5.5]
```

Requirements

- Maintain a keyword heap
- Heap order: n.count >= n.parent.count (MIN-HEAP)

- For the heap structure, you can
  - Use java.util.ArrayList
  - java.util.PriorityQueue
  - Or develop it by yourself

I/O Example: Add

- Input:
  - Token1 : a constant "add"
  - Token2 : keyword name **k**
  - Token3: keyword count **c**
  - Token4: keyword weight **w**
  - **EX:**

```
add Fang 3 0.5
```

- **To do:** Insert a keyword **[k,c,w]** to the heap
- **Output:**
  - Simply output a line of constant "**Done**"
  - EX:

```
Done
```

I/O Example: peek

- Input:
  - Token1 : a constant “peek”
  - EX:

```
peek
```

- **To do:** Output the keyword with minimal count
- **Output:**
  - If heap is empty, simply output a line of “InvalidOperation”:

```
InvalidOperation
```

  - If it is legal to peek:

```
[NCCU,4,9.9]
```

I/O Example: removeMin

- Input:
  - Token1 : a constant "removeMin"
  - EX: `removeMin`
- **To do: Output and Remove** the keyword of the root
- **Output:**
  - If heap is empty, then output "InvalidOperation":

    `InvalidOperation`
  - If it is legal to remove:

    `[NCCU,4,9.9]`

I/O Example: output

- Input:
  - Token1 : a constant “output”
  - EX:

```
output
```

- **To do: Output and Remove** all the keywords in order (ascending)
- **Output:**
  - If heap is empty, then print a empty line:

```
InvalidOperation
```

  - If heap is not empty:

```
[NCCU,4,9.9] [MIS,5,9.9] [DS,6,9.9]
```

Sample Test Data (Input)

- add Fang 3 1.2
- add Yu 5 1.8
- add NCCU 2 0.6
- add UCSB 1 11.9
- peek
- add MIS 4 2.2
- removeMin
- add Badminton 5 0.6
- output

Sample Test Data (Output)

- Done
- Done
- Done
- Done
- [UCSB,1,11.9]
- Done
- [UCSB,1,11.9]
- Done
- [NCCU,2,0.6] [Fang,3,1.2] [MIS,4,2.2] [Badminton,5,0.6] [Yu,5,1.8]