

## Softimize Basic R&D Exercise – Entity Collection

### Guidelines

- You may implement your solution in Java/C#.
- Send your solution to [jobs@softimize.co](mailto:jobs@softimize.co) (pay attention: our domain is **NOT** .com!)
  - Email subject should be as follows: "<<Full Name>> - Basic R&D Exercise"
- If you should have any questions please send them to [jobs@softimize.co](mailto:jobs@softimize.co)

### Requirements

1. You are asked to implement the EntityCollection interface which is specified in the attached Java file.
2. Your implementation should support the following operations:
  - a. Add - adds the entity which is given as input to the collection.
  - b. Remove Max Value - removes the entity with the maximal value from the collection and returns it.
3. You should provide 3 implementations for the following use-cases (A-C), according to the frequencies of performing Add & Remove Max Value in these use-cases:

| <i>Use-Case/<br/>Operation</i>        | <i>A</i> | <i>B</i> | <i>C</i> |
|---------------------------------------|----------|----------|----------|
| <b>Add frequency</b>                  | High     | Medium   | Low      |
| <b>Remove Max Value<br/>frequency</b> | Low      | Medium   | High     |

Each use-case implementation should be optimized in terms of its **WC time complexity** -

- If one operation is more frequent than the other operation (e.g. high vs. low) – then the frequent operation should have the **lowest possible** complexity, whereas the other operation may have higher complexity but still optimized as much as possible.
- If both operations are equally frequent (e.g. medium vs. medium) – then both should have **similar** complexity, which is as low as possible in each operation while taking into account also the need for the same complexity in the other operation.

**Good luck!** 😊