**Algorithm depiction of Merge Sort**

* Let’s say we have an array:

1. Split array into half

(low + high)/2 = (0+6)//2 = 3 (floor division)

1. Again do the same for all unless and until we get the smallest chunks

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2  0 | 4  1 | 6  2 | 1  3 | 3  4 | 7  5 | 5  6 |

Low High

|  |  |  |  |
| --- | --- | --- | --- |
| 2  0 | 4  1 | 6  2 | 1  3 |

|  |  |  |
| --- | --- | --- |
| 3  0 | 7  1 | 5  2 |

|  |  |
| --- | --- |
| 2  0 | 4  1 |

|  |  |
| --- | --- |
| 6  0 | 1  1 |

|  |  |
| --- | --- |
| 3  0 | 7  1 |

|  |
| --- |
| 5  0 |

|  |
| --- |
| 1 |

|  |
| --- |
| 5 |

|  |
| --- |
| 2 |

|  |
| --- |
| 4 |

|  |
| --- |
| 6 |

|  |
| --- |
| 3 |

|  |
| --- |
| 7 |

|  |  |
| --- | --- |
| 3 | 7 |

|  |
| --- |
| 5 |

1. Compare the items and merge into the sorted manner

|  |
| --- |
| 2 |

|  |
| --- |
| 4 |

|  |  |
| --- | --- |
| 1 | 6 |

1. Again compare the items and sort it the same manner

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 4 | 6 |

|  |  |  |
| --- | --- | --- |
| 3 | 5 | 7 |

1. For the last time, Compare the items

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Hence, we’ll be getting an ordered array of elements.