Merge sort uses divide-and-conquer:

1. **Divide** by finding the number q*q*q of the position midway between p*p*p and r*r*r. Do this step the same way we found the midpoint in binary search: add p*p*p and r*r*r, divide by 2, and round down.
2. **Conquer** by recursively sorting the subarrays in each of the two subproblems created by the divide step. That is, recursively sort the subarray array[p..q] and recursively sort the subarray array[q+1..r].
3. **Combine** by merging the two sorted subarrays back into the single sorted subarray array[p..r].

3,4,6,1,8,2

divide

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | 4 | 6 | 1 | 8 | 2 |

divide

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 6 |

|  |  |  |
| --- | --- | --- |
| 1 | 8 | 2 |

|  |
| --- |
| 6 |

|  |
| --- |
| 2 |

divide

|  |  |
| --- | --- |
| 3 | 4 |

|  |  |
| --- | --- |
| 1 | 8 |

|  |
| --- |
| 1 |

|  |
| --- |
| 8 |

|  |
| --- |
| 2 |

|  |
| --- |
| 6 |

|  |
| --- |
| 4 |

divide

|  |
| --- |
| 3 |

|  |  |
| --- | --- |
| 1 | 8 |

|  |
| --- |
| 2 |

merge

|  |  |
| --- | --- |
| 3 | 4 |

|  |
| --- |
| 6 |

merge

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 6 |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 8 |

merge

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 6 | 8 |