


# Binary Search Algorithm

# Binary Search Algorithm

- The array needs to be **sorted** before binary search
- **Step 1:** Start from the center of the array and start to search. If there is no exact center, take the one from the left.
- **Step 2:** If the item we want to search is in the center, the item is found. And the search ends here.
- **Step 3:** If the item to search is greater than the current item, discard the left side of the array and start from step 1 again
- **Step 4:** If the item to search is smaller than the current item, discard the right side of the array and start from step 1 again.

# Binary Search

- The array needs to be sorted (by name) before we can perform the binary search

| Name | Edward | James | John | May | Peter |
|---|---|---|---|---|---|
| Score | 67 | 99 | 57 | 88 | 78 |

# Binary Search

- Search for the score of James.
- Start to search from the centre of the array (which is John)

| Name | Edward | James | John | May | Peter |
|---|---|---|---|---|---|
| Score | 67 | 99 | 57 | 88 | 78 |

- "James" is smaller than "John", **"discard"** the array on the right.

# Binary Search

- **"discard"** does not mean delete the data away. It just means we can ignore that part of the array since the item we want to look for is not in there.

| Name | Edward | James | John | May | Peter |
|---|---|---|---|---|---|
| Score | 67 | 99 | 57 | 88 | 78 |

# Binary Search

- Start to search from the centre of the remaining array. No exact centre, take the one on the left. (i.e. Edward).

| Name | Edward | James | John | May | Peter |
|---|---|---|---|---|---|
| Score | 67 | 99 | 57 | 88 | 78 |

# Binary Search

- Since "James" is greater than "Edward", discard the array on the left.

| Name | Edward | James | John | May | Peter |
|---|---|---|---|---|---|
| Score | 67 | 99 | 57 | 88 | 78 |

- Start to search from the centre of the remaining array.
- The centre name is the name to search
- Name found and process stop.