How to Search!

Something useful to do with data
Goal: Find a number!

- Remember: Computer can only look at two things at a time

What we want: 42
Goal: Find a number!

• Remember: Computer can only look at two things at a time

\[ 5 \ ? \ ? \ ? \ ? \ ? \]

What we want: 42
Goal: Find a number!

- Remember: Computer can only look at two things at a time

What we want: 42
Goal: Find a number!

- Remember: Computer can only look at two things at a time

What we want: 42
Goal: Find a number!

- Remember: Computer can only look at two things at a time

What we want: 42

Success!
Sequential Search

• Search through data items one after another

• Also called Linear Search

• Best search technique when data is out of order (unsorted)
int[] array = ...
int target = ...;
int index = -1;

for(int i = 0; i < array.length; i++)
    if(array[i] == target)
    {
        index = i;
        break;
    }

//here index is set correctly

Given an array with values and something to look for

Not Found
index == -1

If we find our target value, end the loop
Code v.2

finds the position of a target value within the array, returns -1 if the target is not found.

```java
public int search(int[] array, int target) {
    for(int i = 0; i < array.length; i++)
        if(array[i] == target)
            return i;
    return -1;
}
```
public boolean contains(int[] array, int target) {
    for(int i = 0; i < array.length; i++)
        if(array[i] == target)
            return true;
    return false;
}
### Speed Analysis

| 4 | 0 | -5 | 6 | 92 | -4 |

If right at the beginning:

If right at the end:

If in the middle:

On average:
### Speed Analysis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>-5</th>
<th></th>
<th>6</th>
<th>92</th>
<th>-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- If right at the beginning: I check
- If right at the end:
- If in the middle:
- On average:
Speed Analysis

| 4 | 0 | -5 | 6 | 92 | -4 |

If right at the beginning: 1 check
If right at the end: array.length checks
If in the middle:

On average:
Speed Analysis

| 4 | 0 | -5 | 6 | 92 | -4 |

If right at the beginning: 1 check
If right at the end: array.length checks
If in the middle: array.length/2 checks

On average:
Speed Analysis

| 4 | 0 | -5 | 6 | 92 | -4 |

If right at the beginning: 1 check
If right at the end: array.length checks
If in the middle: array.length/2 checks
On average: You have to check half the values in the array
public class Time
{
    private int hours;
    private int minutes;

    public int getHours() {
        return hours;
    }

    public int getMinute() {
        return minutes;
    }
}

Assume military time
Searching with Objects

How to determine which time is the latest?
public static void main(String[] args) {
    Time[] array = ...;
    Time latest = array[0];
    int totalMinutes = latest.getHours() * 60 + latest.getMinutes();
    for(int i = 1; i < array.length; i++) {
        Time current = array[i];
        int currentMinutes = current.getHours() * 60 + current.getMinutes();
        if(currentMinutes < totalMinutes) {
            totalMinutes = currentMinutes;
            latest = current;
        }
    }
    //latest will be assigned the right time
}
Faster Search

• There is a faster way to search if one condition is met: the data is sorted

• Dictionary Example
  • How to search a dictionary?
  • How many page-checks to find a word?
Binary Search

• Idea
  • Take sorted data
  • Repeatedly split into smaller groups
  • Divide and Conquer strategy!
  • Much faster than Linear Search
Binary Search #1

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

Wednesday, February 15, 12
Binary Search #1

-5  -2  0   3   7   10  12  21  43

target = 21
Binary Search #1

lo: 0

target = 21
Binary Search #1

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

lo: 0  hi: 8

target = 21
## Binary Search #1

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

- **lo**: 0
- **hi**: 8

- **target** = 21
- **mid** = \((lo + hi)/2\)
Binary Search #1

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

lo: 0

mid = 4

target = 21

mid = (lo + hi)/2
## Binary Search #1

<table>
<thead>
<tr>
<th>-5</th>
<th>-2</th>
<th>0</th>
<th>3</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>21</th>
<th>43</th>
</tr>
</thead>
</table>

mid = \frac{(lo + hi)}{2}

**target** = 21

mid = (lo + hi)/2

lo = mid + 1
### Binary Search #1

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

- \text{target} = 21
- \text{mid} = (\text{lo} + \text{hi})/2
Binary Search #1

-5  -2  0   3   7   10  12  21  43

lo: 5

target = 21

mid = (lo + hi)/2
Binary Search #1

| -5 | -2 | 0  | 3  | 7  | 10 | 12 | 21 | 43 |

target = 21

mid = (lo + hi)/2

lo: 5
hi: 8
Binary Search #1

mid = (lo + hi)/2

target = 21

mid = (5 + 8)/2 = 6
Binary Search #1

mid = (lo + hi)/2

target = 21
mid = (lo + hi)/2
lo = mid + 1
Binary Search #1

| -5 | -2 | 0  | 3  | 7  | 10 | 12 | 21 | 43 |

target = 21

mid = (lo + hi)/2
target = 21

mid = (lo + hi)/2
### Binary Search #1

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

**target** = 21

**mid** = (lo + hi)/2

lo: 7  hi: 8
Binary Search #1

mid = (lo + hi)/2

lo: 7    hi: 8

mid = 7

target = 21

mid = (lo + hi)/2
Binary Search #1

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

mid = (lo + hi)/2

lo: 7  hi: 8

mid = 7

target = 21

Success!
# Binary Search #2

<table>
<thead>
<tr>
<th>-5</th>
<th>-2</th>
<th>0</th>
<th>3</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>21</th>
<th>43</th>
</tr>
</thead>
</table>
**Binary Search #2**

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

**target = 3**
Binary Search #2

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

lo: 0

target = 3
Binary Search #2

|   -5   |   -2   |    0    |    3    |    7    |   10   |   12   |   21   |   43   |

lo: 0  hi: 8

target = 3
Binary Search #2

mid = (lo + hi)/2

lo: 0

hi: 8

target = 3

mid = (lo + hi)/2
Binary Search #2

mid = (lo + hi)/2

lo: 0

hi: 8

target = 3

mid = 4

mid = (lo + hi)/2
Binary Search #2

mid = (lo + hi)/2

target = 3

lo: 0
hi: 8

mid = 4

hi = mid - 1
### Binary Search #2

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

- **target = 3**
- **mid = (lo + hi)/2**
### Binary Search #2

|   -5  |  -2  |   0  |   3  |   7  |  10  |  12  |  21  |  43  |

- **lo**: 0

- **target** = 3

- **mid** = \((lo + hi)/2\)
| -5 | -2 | 0  | 3  | 7  | 10 | 12 | 21 | 43 |

mid = (lo + hi)/2

lo: 0  hi: 3

target = 3

mid = (lo + hi)/2
**Binary Search #2**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
</tr>
</tbody>
</table>

```
mid = (lo + hi)/2
```

- **lo:** 0
- **hi:** 3

**target = 3**

**mid = (lo + hi)/2**
Binary Search #2

mid = (lo + hi) / 2

target = 3

lo: 0  hi: 3
mid = 1

lo = mid + 1
**Binary Search #2**

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

**target** = 3

**mid** = (lo + hi)/2
## Binary Search #2

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

mid = \( \frac{lo + hi}{2} \)

target = 3

mid = (lo + hi)/2
Binary Search #2

target = 3
mid = (lo + hi)/2
Binary Search #2

mid = (lo + hi)/2

lo: 2  hi: 3

mid = 2

target = 3

mid = (lo + hi)/2
Binary Search #2

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

\[
\text{lo: 2} \quad \text{hi: 3}
\]

mid = 2

\[
\text{target = 3}
\]

mid = (lo + hi)/2

lo = mid + 1
Binary Search #2

-5  -2  0  3  7  10  12  21  43

target = 3

mid = (lo + hi)/2
Binary Search #2

| -5 | -2 | 0  | 3  | 7  | 10 | 12 | 21 | 43 |

lo: 3

target = 3

mid = (lo + hi)/2
Binary Search #2

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

mid = \frac{(lo + hi)}{2}

<table>
<thead>
<tr>
<th>lo: 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>hi: 3</td>
</tr>
</tbody>
</table>

target = 3

mid = (lo + hi)/2
Binary Search #2

\[
\begin{array}{cccccccc}
-5 & -2 & 0 & 3 & 7 & 10 & 12 & 21 & 43 \\
\end{array}
\]

\[\text{lo: 3}\]

\[\text{hi: 3}\]

\[\text{mid} = 3\]

\[\text{target} = 3\]

\[\text{mid} = (\text{lo} + \text{hi})/2\]
Binary Search #2

\[
\text{mid} = \frac{\text{lo} + \text{hi}}{2}
\]

mid = 3

lo: 3

hi: 3

target = 3

Success!

mid = (lo + hi)/2

Wednesday, February 15, 12
## Binary Search #3

<table>
<thead>
<tr>
<th>-5</th>
<th>-2</th>
<th>0</th>
<th>3</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>21</th>
<th>43</th>
</tr>
</thead>
</table>

Wednesday, February 15, 12
target = 44
## Binary Search #3

<table>
<thead>
<tr>
<th>-5</th>
<th>-2</th>
<th>0</th>
<th>3</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>21</th>
<th>43</th>
</tr>
</thead>
</table>

**lo:** 0

**target = 44**
Binary Search #3

|   -5  |  -2  |   0  |   3  |   7  |  10  |  12  |  21  |   43 |

lo: 0  hi: 8

target = 44
Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

lo: 0

hi: 8

target = 44

mid = (lo + hi)/2
Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

\[ \text{mid} = \frac{\text{lo} + \text{hi}}{2} \]

\( \text{target} = 44 \)
\( \text{mid} = \frac{\text{lo} + \text{hi}}{2} \)

\( \text{lo} = 0 \)
\( \text{hi} = 8 \)
### Binary Search #3

<table>
<thead>
<tr>
<th>-5</th>
<th>-2</th>
<th>0</th>
<th>3</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>21</th>
<th>43</th>
</tr>
</thead>
</table>

- **lo**: 0
- **hi**: 8
- **mid**: 4
- **target**: 44

**Formulae**

- mid = (lo + hi) / 2
- lo = mid + 1
**Binary Search #3**

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

target = 44

mid = (lo + hi)/2
Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

lo: 5

target = 44

mid = (lo + hi)/2
target = 44

mid = (lo + hi)/2
Binary Search #3

mid = (lo + hi)/2

target = 44

mid = 6

lo: 5

hi: 8

-5  -2  0  3  7  10  12  21  43
Binary Search #3

mid = (lo + hi)/2

lo: 5

hi: 8

mid = 6

target = 44

lo = mid + 1

Wednesday, February 15, 12
Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

`target = 44`

`mid = (lo + hi)/2`
### Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

```plaintext
mid = (lo + hi)/2

\[
\text{target} = 44
\]

\[
\text{mid} = (\text{lo} + \text{hi})/2
\]
```
Binary Search #3

\[
\begin{array}{cccccccc}
-5 & -2 & 0 & 3 & 7 & 10 & 12 & 21 & 43 \\
\end{array}
\]

\[
\text{target} = 44
\]

\[
\text{mid} = \frac{(\text{lo} + \text{hi})}{2}
\]
# Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

Thursday, February 15, 12

\[
\text{target} = 44
\]

\[
\text{mid} = \frac{(\text{lo} + \text{hi})}{2}
\]

lo: 7  
hi: 8  
mid = 7
Binary Search #3

| -5 | -2 | 0  | 3  | 7  | 10 | 12 | 21 | 43 |

mid = (lo + hi)/2

**target = 44**

mid = (lo + hi)/2

lo = mid + 1

lo: 7  hi: 8

mid = 7
Binary Search #3

\[
\begin{array}{cccccccc}
-5 & -2 & 0 & 3 & 7 & 10 & 12 & 21 & 43 \\
\end{array}
\]

target = 44

mid = (lo + hi)/2
target = 44
mid = (lo + hi)/2
Binary Search #3

-5  -2  0  3  7  10  12  21  43

target = 44

mid = (lo + hi)/2

lo: 8
hi: 8
### Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

- \( \text{target} = 44 \)
- \( \text{mid} = \frac{\text{lo} + \text{hi}}{2} \)
- \( \text{hi: 8} \)
- \( \text{lo: 8} \)
- \( \text{mid = 8} \)
Binary Search #3

mid = (lo + hi)/2

hi: 8
lo: 8
mid = 8

target = 44

lo = mid + 1

Wednesday, February 15, 12
### Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

**target** = 44

**mid** = (lo + hi)/2
### Binary Search #3

| -5 | -2 | 0 | 3 | 7 | 10 | 12 | 21 | 43 |

- **target** = 44
- **mid** = \((\text{lo} + \text{hi})/2\)

**lo**: 9
### Binary Search #3

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

- **target** = 44  
- **mid** = \((\text{lo} + \text{hi})/2\)

**lo:** 9  
**hi:** 8
## Binary Search #3

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>21</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

**target** = 44  
**mid** = \((lo + hi)/2\)

*Failure!*

hi: 8  
lo: 9

*Wednesday, February 15, 12*
```java
public int binarySearch(int[] array, int target) {
    int lo = 0;
    int hi = array.length - 1;

    while (lo <= hi) {
        int mid = (lo + hi) / 2;
        if (array[mid] == target)
            return mid;
        else if (array[mid] < target)
            lo = mid + 1;
        else
            hi = mid - 1;
    }

    return -1;
}
```
public int binarySearch(int[] array, int target) {
    return binarySearch(array, target, 0, array.length-1);
}

public int binarySearch(int[] array, int target, int lo, int hi) {
    if(lo > hi)
        return -1;
    int mid = (lo + hi)/2;
    if(array[mid] == target)
        return mid;
    else if(array[mid] < target)
        return binarySearch(array, target, mid + 1, hi);
    else
        return binarySearch(array, target, lo, mid - 1);
}
Speed Analysis

- Binary Search is repeated division by 2
- The question is, how many times do you have to divide by 2 (at worst) to find what you're looking for?
- Repeated division by 2 is the same as $\log_2$
- $\log_2(16) = 4$
## Comparison

<table>
<thead>
<tr>
<th>array.length</th>
<th>Sequential</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1,024</td>
<td>512</td>
<td>10</td>
</tr>
<tr>
<td>1,048,576</td>
<td>524,288</td>
<td>20</td>
</tr>
</tbody>
</table>

n / 2

\[ \log_2(n) \]