CSCI 241
Scott Wehrwein
Algorithm Case Study: Binary Search
Goals

Understand how the binary search algorithm works, and under what conditions.

Be able to execute binary search on paper.
Searching an array

Goal: return the index of \( v \) in \( A \).

The straightforward way:

```plaintext
for i = 0..A.length:
    if A[i] == v:
        return i
return -1
```
Searching an array

Goal: return the index of v in A.

The straightforward way:

```java
for i = 0..A.length:
    if A[i] == v:
        return i
return -1
```

Can we do better?
Searching an array

Goal: return the index of $v$ in $A$.

The straightforward way:

```java
for i = 0..A.length:
    if A[i] == v:
        return i
return -1
```

Can we do better?

Nope (in general).
Searching a **sorted** array

Goal: return the index of \( v \) in \( A \).
Precondition: \( A \) is sorted.

```
start                     end
A
```
Searching a **sorted** array

Goal: return the index of \( v \) in \( A \).
Precondition: \( A \) is sorted.

\[
\begin{array}{cccc}
\text{start} & \text{mid} & \text{end} \\
A & x & \\
\end{array}
\]
Searching a **sorted** array

Goal: return the index of \( v \) in \( A \).

Precondition: \( A \) is sorted.

\[
\begin{array}{c|c|c}
\text{start} & \text{mid} & \text{end} \\
\hline
A & x & \\
\end{array}
\]

If \( v < x \),
\( v \) can't be after \( \text{mid} \)!
Searching a sorted array

Goal: return the index of $v$ in $A$.
Precondition: $A$ is sorted.

If $v > x$, $v$ can't be before $mid$!
If $v < x$, $v$ can't be after $mid$!
Binary Search: Example

\[
\text{binarySearch}(A, 21)
\]

\[
A: [1, 1, 2, 3, 5, 8, 13, 21, 34]
\]

\[21 > 5\]

\[
[1, 1, 2, 3, 5, 8, 13, 21, 34]
\]

\[U > 13\]

\[
[1, 1, 2, 3, 5, 8, 13, 21, 34]
\]

\[U = 21\]
Binary Search: Example

\( \text{binarySearch}(A, 4) \)

\[
A: \ [1, 1, 2, 3, 5, 8, 13, 21, 34] \\
S \leq 4 <s \\
\leq [1, 1, 2, 3, 5, 8, 13, 21, 34] \\
4 \geq m \rightarrow 4 \geq 3 \leq [1, 1, 2, 3, 5, 8, 13, 21, 34] \\
s < 4 \rightarrow S \text{ empty return}