# MCAC 301: Design and Analysis of Algorithms

Neelima Gupta

ngupta@cs.du.ac.in

April 27, 2023

# Can we search faster?

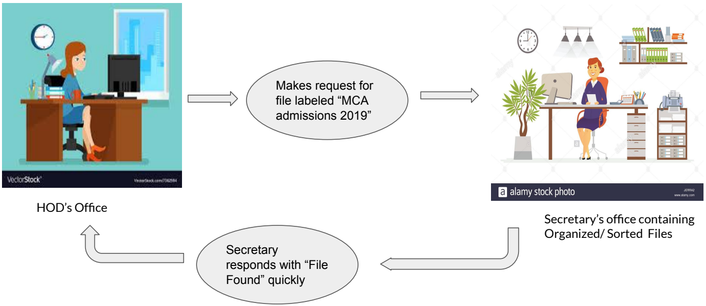



Figure A: Secretary at an office responding to the requests of the Head

1

# Searching for a word in a dictionary

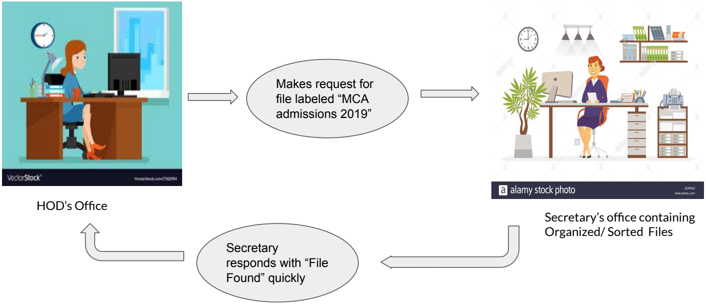
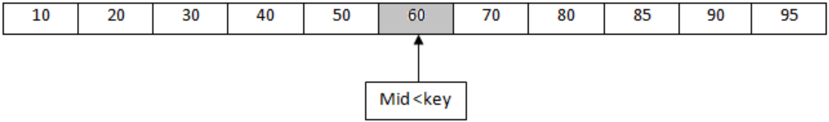



Figure A: Secretary at an office responding to the requests of the Head

1

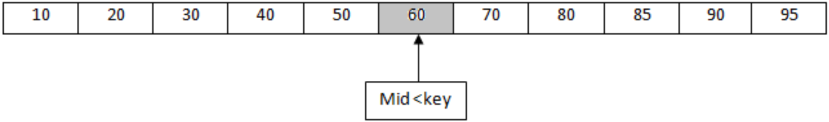
# Binary Search

**Key:** 65



# Binary Search

**Key:** 65



# Divide and Conquer Paradigm

1. Divide : Divide the problem into a number of subproblems of smaller Size.
2. Conquer : Solve each sub-problem.
3. Combine : Combine the solution of the subproblems to obtain the solution of the original (bigger) problem.

A subproblem is solved recursively (i.e. by applying the same algorithm on the smaller problem) so long as it's size $>$ some threshold (called the "terminating condition"). Thereafter, it is solved directly.

# Binary Search: Split in the middle

**input :** Array: $A[1 \dots n]$, $Key$
**output:** Index of key if key found, -1 otherwise

```
Binary-Search-Recursive(A, first, last, key)
/* "first" and "last" are the first and the last indices of the
   array
if first ≤ last then
    mid = (first + last)/2
    if A[mid]=key then
        return mid
    end
    if A[mid] < key then
        Binary-Search-Recursive(A, mid + 1, last, key)
    end
    else
        Binary-Search-Recursive(A, first, mid -1, key)
    end
end
```

# Frame Title

Quick Demo

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,
$W(n) = ?$

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,
$W(n) = ?$
$W(n) = W(n/2) + ?$

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,
$W(n) = ?$
$W(n) = W(n/2) + ?$
$W(n) = W(n/2) + 2$

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,

$W(n) = ?$

$W(n) = W(n/2) + ?$

$W(n) = W(n/2) + 2$

$W(n) = ?$

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,
$W(n) =?$
$W(n) = W(n/2)+?$
$W(n) = W(n/2) + 2$
$W(n) =?$  $W(n) = \log n$
Let $B(n)$ be the number of comparisons performed by the binary search algorithm in best case. Then,

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,
$W(n) = ?$
$W(n) = W(n/2) + ?$
$W(n) = W(n/2) + 2$
$W(n) = ?$ $\quad W(n) = \log n$
Let $B(n)$ be the number of comparisons performed by the binary search algorithm in best case. Then,
$B(n) = ?$

# Time Complexity

Let $W(n)$ be the number of comparisons performed by the binary search algorithm in worst case. Then,

$W(n) =?$

$W(n) = W(n/2)+?$

$W(n) = W(n/2) + 2$

$W(n) =?$ $W(n) = \log n$

Let $B(n)$ be the number of comparisons performed by the binary search algorithm in best case. Then,

$B(n) =?$

$B(n) = 1$