

CHAPTER 10: SEARCHING ALGORITHMS

10.1 LINEAR SEARCH

10.1.1 How It Works

1. Start at first element
2. Compare with target
3. If match, return position
4. If not, move to next element
5. Repeat until found or end of list

10.1.2 Pseudocode

<PSEUDOCODE>

```
DECLARE myList : ARRAY[0:9] OF INTEGER
DECLARE upperBound : INTEGER
DECLARE lowerBound : INTEGER
DECLARE index : INTEGER
DECLARE item : INTEGER
DECLARE found : BOOLEAN

upperBound ← 9
lowerBound ← 0
OUTPUT "Enter item to find"
INPUT item
found ← FALSE
index ← lowerBound

REPEAT
  IF item = myList[index] THEN
    found ← TRUE
  ELSE
    index ← index + 1
```

10.1.3 Python Code

<PYTHON>

```
myList = [4, 2, 8, 17, 9, 3, 7, 12, 34, 21]
item = int(input("Enter item to find: "))
found = False

for x in range(len(myList)):
    if myList[x] == item:
        found = True
        print(f"Item found at index {x}")
        break
```

10.1.4 Complexity

- ```
if not found:
```
- **Time:**  $O(n)$   
`print("Item not found")`
  - **Space:**  $O(1)$
-

## 10.2 BINARY SEARCH

---

### 10.2.1 Prerequisites

- List MUST be sorted

### 10.2.2 How It Works

1. Find middle element
2. Compare with target
3. If match, return position
4. If target > middle, search right half
5. If target < middle, search left half
6. Repeat until found or sublist empty

### 10.2.3 Pseudocode

#### <PSEUDOCODE>

```
DECLARE myList : ARRAY[0:9] OF INTEGER
DECLARE UB : INTEGER
DECLARE LB : INTEGER
DECLARE index : INTEGER
DECLARE item : INTEGER
DECLARE found : BOOLEAN
```

```
UB ← 9
```

```
LB ← 0
```

```
OUTPUT "Enter item to find"
```

```
INPUT item
```

```
found ← FALSE
```

```
REPEAT
```

```
 index ← INT((UB + LB) / 2)
```

```
 IF item = myList[index] THEN
```

```
 found ← TRUE
```

```
 ELSE
```

```
 IF item > myList[index] THEN
```

```
 LB ← index + 1
```

```
 ELSE
```

## 10.2.4 Python Code

<PYTHON>

```
myList = [16, 19, 21, 27, 36, 42, 55, 67, 76]
```

```
item = int(input("Enter item to find: "))
```

```
min = 0
```

```
max = len(myList) - 1
```

```
found = False
```

```
index = 0
```

```
while found is False and min <= max:
```

```
 index = int((max + min) / 2)
```

```
 if item == myList[index]:
```

```
 found = True
```

```
 elif item < myList[index]:
```

```
 max = index - 1
```

## 10.2.5 Complexity

- **Time:**  $O(\log n)$
- **Space:**  $O(1)$

---

Revision #1

Created 2026-03-16 12:18:22 UTC by Samuel Lee

Updated 2026-03-16 12:18:40 UTC by Samuel Lee