

# Intermediate C Programming

## Lesson6

Search (1)

---

# Today's outline

- Search

- Serial search

- Exercise

---

---

# ■ Search

- Search

Searching Algorithm is to find an item with specified properties among a collection of items

To find the specified item from a huge number of items

- Algorithms

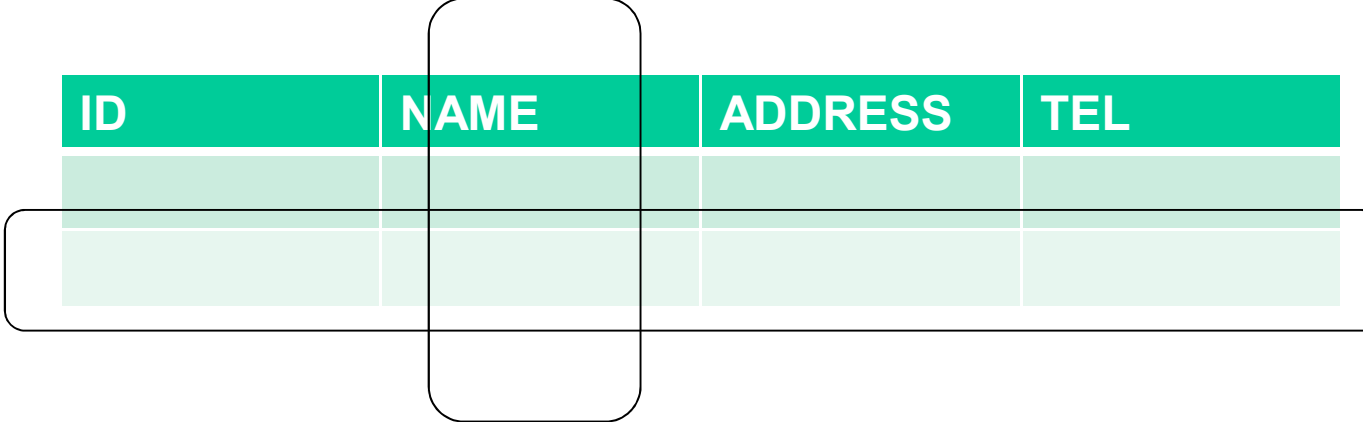
1. Serial search
  2. Binary search
-

# ■ Search

*Column*

ID	NAME	ADDRESS	TEL

*Record*

A diagram of a database table. The table has four columns: ID, NAME, ADDRESS, and TEL. The first row is the header row, highlighted in green. Below it are two empty rows. A vertical rounded rectangle highlights the NAME column, with the label 'Column' above it. A horizontal rounded rectangle highlights the first empty row, with the label 'Record' to its left.

- Record

One person's data

- Column

Properties such as name ,add, tel

- Key

Record to search

---

# ■ Search and Cost

- Cost 1

Cost of searching

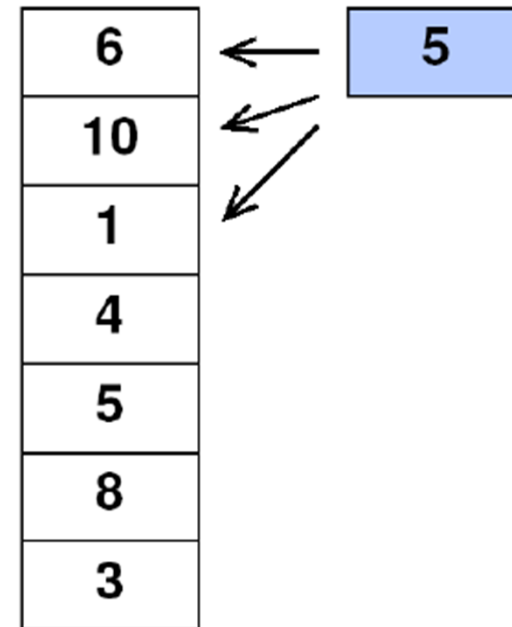
- Cost 2

Cost of adding/removing data

---

# ■ Serial Search

- Search from the beginning of items
- No guarantee that the target must be available in the list
- Best case: find the wanted item by search once
- Worst case: find the wanted data after searching all items
- Average:  $(n+1)/2$



---

# ■ Serial Search

- Search with the unsorted item

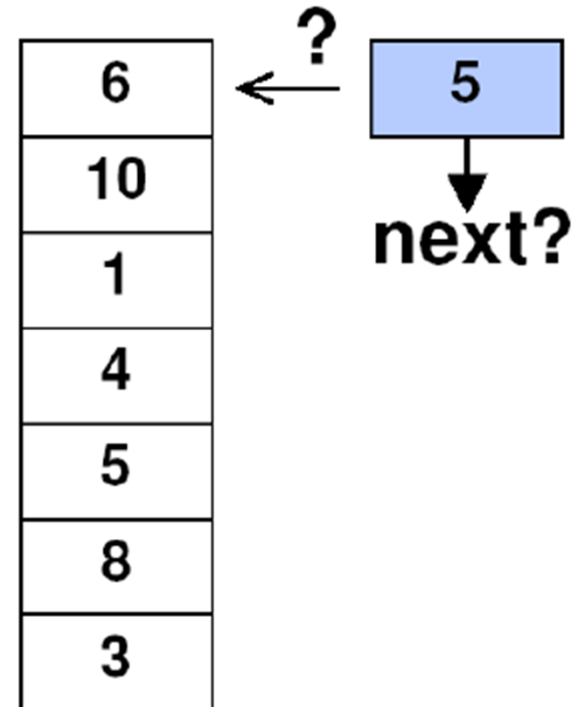
```
#define N 7  
int data[N] = { 6, 10, 1, 4, 5, 8, 3 };
```

# ■ Serial Search

- Search from the 1<sup>st</sup> item
- Cost

Is it the same as the key?  
Is it the last item of the list

- For one item, twice of comparison





---

# ■ Serial Search

```
for( i=0; i<n; i++ ){  
    if( data[i] == key ){  
        break;  
    }  
}
```

# Exercise

```
#include <stdio.h>
#define N 1000000
int main() {
    FILE *datafile;
    int i, size, key, data[N];
    char filename[20];
    printf( "data file: " );
    scanf( "%s", filename );
    datafile = fopen( filename, "r" );
    for( i=0; i<N; i++ ){
        if( fscanf(datafile, "%d", &data[i]) == EOF ){
            break;
        }
    }
    size = i;

    return 0;
}
```