Fast campus

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 32강 - 순차 탐색과 이진 탐색

Fast campus

학습 목표

순차 탐색과 이진 탐색

1) 순차 탐색과 이진 탐색의 원리를 이해하고 이를 C언어로 구현할 수 있습니다.

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 특정한 원소를 찾기 위해 원소를 순차적으로 하나씩 탐색하는 방법입니다.

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 특정한 원소를 찾기 위해 원소를 순차적으로 하나씩 탐색하는 방법입니다.

찾을 문자열	강종구

인덱스	0	1	2	3	4
원소	나동빈	이태일	박한울	강종구	이상욱

Fast campus

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 특정한 원소를 찾기 위해 원소를 순차적으로 하나씩 탐색하는 방법입니다.

찾을 문자열	강종구

인덱스	0	1	2	3	4
원소	나동빈	이태일	박한울	강종구	이상욱

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 특정한 원소를 찾기 위해 원소를 순차적으로 하나씩 탐색하는 방법입니다.

찾을 문자열	강종구

인덱스	0	1	2	3	4
원소	나동빈	이태일	박한울	강종구	이상욱

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 특정한 원소를 찾기 위해 원소를 순차적으로 하나씩 탐색하는 방법입니다.

찾을 문자열	강종구

인덱스	0	1	2	3	4
원소	나동빈	이태일	박한울	강종구	이상욱

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 특정한 원소를 찾기 위해 원소를 순차적으로 하나씩 탐색하는 방법입니다.

찾을 문자열	강종구

인덱스	0	1	2	3	4
원소	나동빈	이태일	박한울	강종구	이상욱

순차 탐색과 이진 탐색

문자열 순차 탐색 구현 1)

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define LENGTH 1000

char **array;
int founded;
```

순차 탐색과 이진 탐색

문자열 순차 탐색 구현 2)

```
int main(void) {
  int n;
  char *word;
  word = malloc(sizeof(char) * LENGTH);
  scanf("%d %s", &n, word);
  array = (char**) malloc(sizeof(char*) * n);
  for (int i = 0; i < n; i++) {
    array[i] = malloc(sizeof(char) * LENGTH);
    scanf("%s", array[i]);
  }
  for (int i = 0; i < n; i++) {
    if (!strcmp(array[i], word)) {
      printf("%d번째 원소입니다.\n", i + 1);
      founded = 1;
    }
  }
  if (!founded) printf("원소를 찾을 수 없습니다.\n");
  system("pause");
  return 0;
}
```

순차 탐색과 이진 탐색

순차 탐색

순차 탐색(Sequential Search)은 데이터 정렬 유무에 상관 없이 가장 앞에 있는 원소부터 하나씩 확인해야 한다는 점에서 $O(N)$의 시간 복잡도를 가집니다.

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)은 배열 내부 데이터가 이미 정렬 되어 있는 상황에서 사용 가능한 알고리즘입니다. 탐색 범위를 절반씩 좁혀가며 데이터를 탐색하는 특징이 있습니다.

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)은 배열 내부 데이터가 이미 정렬 되어 있는 상황에서 사용 가능한 알고리즘입니다. 탐색 범위를 절반씩 좁혀가며 데이터를 탐색하는 특징이 있습니다.

찾을 원소	37

인덱스	0	1	2	3	4	5	6	7	8
원소	15	27	37	46	57	69	73	85	98

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)은 배열 내부 데이터가 이미 정렬 되어 있는 상황에서 사용 가능한 알고리즘입니다. 탐색 범위를 절반씩 좁혀가며 데이터를 탐색하는 특징이 있습니다.

찾을 원소	37

인덱스	0	1	2	3	4	5	6	7	8
원소	15	27	37	46	57	69	73	85	98

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)은 배열 내부 데이터가 이미 정렬 되어 있는 상황에서 사용 가능한 알고리즘입니다. 탐색 범위를 절반씩 좁혀가며 데이터를 탐색하는 특징이 있습니다.

찾을 원소	37

인덱스	0	1	2	3	4	5	6	7	8
원소	15	27	37	46	57	69	73	85	98

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)은 배열 내부 데이터가 이미 정렬 되어 있는 상황에서 사용 가능한 알고리즘입니다. 탐색 범위를 절반씩 좁혀가며 데이터를 탐색하는 특징이 있습니다.

찾을 원소	37

인덱스	0	1	2	3	4	5	6	7	8
원소	15	27	37	46	57	69	73	85	98

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)는 한 번 확인할 때마다 보아야 하는 원소의 개수가 절반씩 줄어든다는 점에서 탐색 시간이 $O(logN)$의 시간 복잡도를 가집니다.

찾을 원소	37

MID 위치에 있는 원소와 반복적으로 비교

인덱스	0	1	2	3	4	5	6	7	8
원소	15	27	37	46	57	69	73	85	98

START MID END

순차 탐색과 이진 탐색

이진 탐색 정의

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define MAX_SIZE 100000

int a[MAX_SIZE];
int founded = 0;

int search(int start, int end, int target) {
  if (start > end) return -9999;
  int mid = (start + end) / 2;
  if (a[mid] == target) return mid;
  else if (a[mid] > target) return search(start, mid - 1, target);
  else return search(mid + 1, end, target);
}
```

순차 탐색과 이진 탐색

이진 탐색 정의

```c
int main(void) {
  int n, target;
  scanf("%d %d", &n, &target);
  for (int i = 0; i < n; i++) scanf("%d", &a[i]);
  int result = search(0, n - 1, target);
  if (result != -9999) printf("%d번째 원소입니다.\n", result + 1);
  else printf("원소를 찾을 수 없습니다.\n");
  system("pause");
  return 0;
}
```

순차 탐색과 이진 탐색

이진 탐색

이진 탐색(Binary Search)는 한 번 확인할 때마다 보아야 하는 원소의 개수가 절반씩 줄어든다는 점에서 탐색 시간에서 $O(logN)$의 시간 복잡도를 가집니다.

배운 내용 정리하기

순차 탐색과 이진 탐색

1) 순차 탐색은 $O(N)$의 시간 복잡도를 가집니다.
2) 이진 탐색은 정렬이 된 상태에서만 사용 가능하며 $O(logN)$의 시간 복잡도를 가집니다.