

# Binary Search 알고리즘

과목: 자료구조

학과: 항공컴퓨터전공

학년: 2

학번: 202200654

이름: 김효준

수강요일: 월

제출일: 2023-04-02

1번.

```
#include<stdio.h> //표준 입출력 함수를 사용할 수 있게하는 헤더파일
#include<stdlib.h> //rand() 함수를 사용할 수 있게 하는 헤더파일
#include<time.h> //시간 관련 헤더파일
#include<stdbool.h> //boolean type 자료형을 사용할 수 있도록 하는 헤더파일
#define MAX_SIZE 101 //배열의 최대 길이

//함수 미리 선언
void sort(int[], int);
void swap(int* n1, int* n2);
int binSearch(int list[], int arrlen, int target);

void main(void){
    int i, n; //i: 접근할 element의 index ; n: list의 길이
    int list[MAX_SIZE]; //정렬할 배열
    srand((unsigned int)time(NULL)); //rand()함수가 시간에 따라 랜덤한 값을 가지도록 하는 함수.

    //배열의 길이를 입력받음
    printf("Enter the number of numbers to generate: ");
    scanf("%d", &n);

    //n이 허용 가능한 범위를 넘어섰을 때 오류처리
    if(n<1 || n>MAX_SIZE){
        fprintf(stderr, "Improper value of n\n");
        exit(1);
    }

    //n개의 element에 각각 랜덤한 값을 삽입
    for(i=0;i<n;i++){
        list[i]=rand() % 1000;
    }

    sort(list, n); //리스트 정렬

    //정렬된 리스트를 출력
    printf("list : ");
    for(i=0;i<n;i++){
        printf("%d ", list[i]);
    }
    printf("\n");

    int target = list[rand()%n]; //찾을 목표 설정
    //찾은 목표가 있는 위치 출력
    printf("target %d는 %d번째 element에 있습니다.", target, binSearch(list, n, target)+1);
    return 0;
}

//매크로 함수인 SWAP을 일반 함수로 바꾼 swap
void swap(int * n1, int * n2){ //두 변수의 주소를 입력받음
    int tmp = *n1;
    *n1=*n2;
    *n2=tmp;
}

//sort함수 정의
void sort(int list[], int n){ //list: 정렬 대상인 배열 ; n: list의 길이
    int i, j, min;
    //i: Access하고 있는 element의 index.
    //j: list[i]와 비교하고 있는 element의 index.
    //min: i번째 작은 수의 index

    for(i=0;i<n-1;i++){
        min=i; //먼저 index가 i인 element를 최솟값으로 설정한다.
        //index가 i+1인 element부터 차례로 비교해간다.
        for(j=i+1;j<n;j++){
            //list[j]가 list[min]보다 작으면 min에 j를 대입한다.
            if(list[j]<list[min])
                min=j;
        }
        swap(&list[i], &list[min]); //i번째로 작은 값을 가진 element의 값과 index가 i인 element의 값을 서로 바꾼다.
    }
}
```

```

//Binary Search
/* parameter
list: target이 있을 것으로 추정되는 list
arrlen: list의 길이
target: 찾고자 하는 대상
*/
/* return: target이 있는 element의 index */
int binSearch(int list[], int arrlen, int target){
    int left = 0, right = arrlen-1; //left는 맨 처음 위치, right는 맨 마지막 위치로 설정
    int mid = (left+right)/2; //mid는 left와 right의 중간 위치로 설정
    while(true){
        // target이 mid위치에 있는 element의 값보다 크면 left를 mid 바로 뒤로 바꾸고 다시 반복한다.
        if(target>list[mid])
            left=mid+1;
        //target이 mid위치에 있는 element의 값과 같으면 mid값을 반환한다.
        else if(target==list[mid])
            return mid;
        //target이 mid위치에 있는 element의 값보다 작으면 right를 mid 바로 앞으로 바꾸고 다시 반복
        else
            right=mid-1;
        if(left>right) //left가 right보다 커지면 target이 list에 없거나 list가 정렬되지 않은 것이므로 -1을 반환.
            return -1;
        mid = (left+right)/2;
    }
}

```

```

Enter the number of numbers to generate: 10
list : 45 176 307 329 377 452 514 737 836 910
target 176는 2번째 element에 있습니다.
-----
Process exited after 1.846 seconds with return value 38
계속하려면 아무 키나 누르십시오 . . .

```

## 2번

```

#include<stdio.h> //표준 입출력 함수를 사용할 수 있게하는 헤더파일
#include<stdlib.h> //rand() 함수를 사용할 수 있게 하는 헤더파일
#include<time.h> //시간 관련 헤더파일
#define MAX_SIZE 101 //배열의 최대 길이

//함수 미리 선언
void sort(int[], int);
void swap(int* n1, int* n2);
int binSearch(int list[], int target, int left, int right);

void main(void){
    int i, n; //i: 접근할 element의 index ; n: list의 길이
    int list[MAX_SIZE]; //정렬할 배열
    srand((unsigned int)time(NULL)); //rand()함수가 시간에 따라 랜덤한 값을 가지도록 하는 함수.

    //배열의 길이를 입력받음
    printf("Enter the number of numbers to generate: ");
    scanf("%d", &n);

    //n이 허용 가능한 범위를 넘어섰을 때 오류처리
    if(n<1 || n>MAX_SIZE){

```

```

    fprintf(stderr, "Improper value of n\n");
    exit(1);
}

//n개의 element에 각각 랜덤한 값을 삽입
for(i=0;i<n;i++){
    list[i]=rand() % 1000;
}

sort(list, n); //리스트 정렬

//정렬된 리스트를 출력
printf("list : ");
for(i=0;i<n;i++){
    printf("%d ", list[i]);
}
printf("\n");

int target = list[rand()%n]; //target을 안에 있는 element 중에서 랜덤하게 결정.
//찾은 목표가 있는 위치 출력
printf("target %d는 %d번째 element에 있습니다.", target, binSearch(list, target, 0, n-1)+1);
return 0;
}

//매크로 함수인 SWAP을 일반 함수로 바꾼 swap
void swap(int * n1, int * n2){ //두 변수의 주소를 입력받음
    int tmp = *n1;
    *n1=*n2;
    *n2=tmp;
}

//sort함수 정의
void sort(int list[], int n){ //list: 정렬 대상인 배열 ; n: list의 길이
    int i, j, min;
    //i: Access하고 있는 element의 index.
    //j: list[i]와 비교하고 있는 element의 index.
    //min: i번째 작은 수의 index

    for(i=0;i<n-1;i++){
        min=i; //먼저 index가 i인 element를 최솟값으로 설정한다.
        //index가 i+1인 element부터 차례로 비교해간다.
        for(j=i+1;j<n;j++){
            //list[j]가 list[min]보다 작으면 min에 j를 대입한다.
            if(list[j]<list[min])
                min=j;
        }
        swap(&list[i], &list[min]); //i번째로 작은 값을 가진 element의 값과 index가 i인 element의 값을 서로 바꾼다.
    }
}

//Binary Search
/* parameters
list: target이 있을 것으로 추정되는 list
target: 찾고자 하는 대상
left: target이 있는 위치 범위의 최솟값
right: target이 있는 위치 범위의 최댓값
*/
/* return: target이 있는 element의 index */
int binSearch(int list[], int target, int left, int right){
    int mid = (left+right)/2;
    if(target>list[mid])
        return binSearch(list, target, mid+1, right);
    else if(target==list[mid])
        return mid;
    else
        return binSearch(list, target, left, mid-1);
}

```

```

Enter the number of numbers to generate: 10
list : 73 99 115 156 200 269 413 424 562 655
target 562는 9번째 element에 있습니다.

```

```

-----
Process exited after 2.476 seconds with return value 38
계속하려면 아무 키나 누르십시오 . . .

```