

[R E P O R T]

정보통신공학전공

200301582

김성태

[Binary-Search (이진 검색) 알고리즘 구현 성능평가 보고서]

1. 목적

: Binary-Search 알고리즘을 구현하여 구현된 알고리즘의 성능을 평가한다.

2. 평가 방법

1. 단순 호출 이진 검색 프로그램 작성
2. 재귀적 호출 이진 검색 프로그램 작성
3. 구현한 두 개의 프로그램에 같은 조건을 상정하여 수행시간을 비교.

3. 프로그램 구현

1. 단순 호출 이진 검색 프로그램

```
/* Created by Anjuta version 1.2.4a */
/* This file will not be overwritten */

/*
 * binay-search.c
 * Written by ST. Kim
 */

#include <stdio.h>
#include <time.h>

/* name : binsearch
 * discription : find the index, by binary search
 */
void binsearch(int start, int end, const int *S, int x, int *location) {

    int low, high, mid, count;

    low = start;
    high = end;
    count = 0;
    *location = 0;
    while (low <= high && *location == 0) {

        count++;
        printf("현재 %d번째의 루프를 수행중입니다.\n", count);
        mid = (low + high) / 2;
        if (x == S[mid])
```

```

        *location = mid;
    else if (x < S[mid]) {
        high = mid - 1;
    }
    else {
        low = mid + 1;
    }
}
}

int main()
{
    struct timespec tv1, tv2; // 시간측정을 위한 timespec struct. 아래에 struct 내용이 나와있음.
    /*
    struct timespec {
        time_t tv_sec;    // seconds
        long tv_nsec;    // nanoseconds
    };
    */
    int i, size, searum, index, time_check;

    printf("지금부터 이진검색 알고리즘을 실행합니다.\n\n\n");
    printf("원하시는 배열의 크기를 입력해 주세요.\n");
    printf("배열의 크기 : ");
    scanf("%d", &size);
    size += 1;

    int S[size];

    for (i = 0; i < size; i++) {
        S[i] = i;
    }

    printf("\n찾고자 하시는 숫자를 입력해주세요.\n");
    printf("숫자 입력! : ");
    scanf("%d", &searum);

    printf("\n잠시만 기다려 주세요....\n\n\n");

    time_check = clock_gettime(CLOCK_REALTIME, &tv1); // time-stamp tv1
    binsearch(0, size, S, searum, &index);

    printf("\n\n찾았습니다!!!\n");
    printf("배열의 인덱스의 위치는 %d 입니다.\n\n", index);
}

```

```

time_check= clock_gettime(CLOCK_REALTIME, &tv2); // time-stamp tv2
printf("단순호출 함수 실행 시간입니다.\n\n실행시간 : %ds, %ldns\n", tv2.tv_sec - tv1.tv_sec,
tv2.tv_nsec - tv1.tv_nsec);

return (0);
}

```

2. 재귀적 호출 이진 검색 프로그램

```

/* Created by Anjuta version 1.2.4a */
/* This file will not be overwritten */

/*
 * binay-recursive.c
 * Written by ST. Kim
 */

#include <stdio.h>
#include <time.h>

int recursive_count = 0; // 재귀호출 카운트를 위한 전역 변수

/* name : binsearch
 * discription : find the index, by binary search
 */
void binsearch(int start, int end, const int *S, int x, int *location) {

    int low, high, mid;

    low = start;
    high = end;

    *location = 0;
    while (low <= high && *location == 0) {

        recursive_count++;
        printf("현재 %d번째의 루프를 수행중입니다.\n", recursive_count);
        mid = (low + high) / 2;
        if (x == S[mid])
            *location = mid;
        else if (x < S[mid]) {
            high = mid - 1;
            binsearch(low, high, S, x, location);
        }
        else {

```

```

        low = mid + 1;
        binsearch(low, high, S, x, location);
    }
}

int main()
{
    struct timespec tv1, tv2; // 시간측정을 위한 timespec struct. 아래에 struct 내용이 나와있음.
    /*
    struct timespec {
        time_t tv_sec;    // seconds
        long tv_nsec;    // nanoseconds
    };
    */
    int i, size, searnum, index, time_check;

    printf("지금부터 이진검색 알고리즘을 실행합니다.\n\n\n");
    printf("원하시는 배열의 크기를 입력해 주세요.\n");
    printf("배열의 크기 : ");
    scanf("%d", &size);
    size += 1;

    int S[size];

    for (i = 0; i < size; i++) {
        S[i] = i;
    }

    printf("\n찾고자 하시는 숫자를 입력해주세요.\n");
    printf("숫자 입력! : ");
    scanf("%d", &searnum);

    printf("\n잠시만 기다려 주세요....\n\n\n");

    time_check = clock_gettime(CLOCK_REALTIME, &tv1); // time-stamp tv1

    binsearch(0, size, S, searnum, &index);

    printf("\n\n찾았습니다!!!\n");
    printf("배열의 인덱스의 위치는 %d 입니다.\n\n", index);

    time_check = clock_gettime(CLOCK_REALTIME, &tv2); // time-stamp tv2
    printf("재귀적 함수 실행 시간입니다.\n\n실행시간 : %ds, %ldns\n", tv2.tv_sec - tv1.tv_sec,
    tv2.tv_nsec - tv1.tv_nsec);
}

```

```
return (0);
```

4. 구현된 프로그램 성능 평가

1. 단순 호출 프로그램

```
pchero@MyNote: ~/Desktop/Study/REPORT/안효범 교수님/알고리즘/binay search
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)

원하시는 배열의 크기를 입력해 주세요.
배열의 크기 : 100000

찾고자 하시는 숫자를 입력해 주세요.
숫자 입력! : 1

잠시만 기다려 주세요....

현재 1번째의 루프를 수행중입니다.
현재 2번째의 루프를 수행중입니다.
현재 3번째의 루프를 수행중입니다.
현재 4번째의 루프를 수행중입니다.
현재 5번째의 루프를 수행중입니다.
현재 6번째의 루프를 수행중입니다.
현재 7번째의 루프를 수행중입니다.
현재 8번째의 루프를 수행중입니다.
현재 9번째의 루프를 수행중입니다.
현재 10번째의 루프를 수행중입니다.
현재 11번째의 루프를 수행중입니다.
현재 12번째의 루프를 수행중입니다.
현재 13번째의 루프를 수행중입니다.
현재 14번째의 루프를 수행중입니다.
현재 15번째의 루프를 수행중입니다.
현재 16번째의 루프를 수행중입니다.
현재 17번째의 루프를 수행중입니다.

찾았습니다!!!
배열의 인덱스의 위치는 1 입니다.

단순호출 함수 실행 시간입니다.

실행시간 : 0s, 259810ns
pchero@MyNote:~/Desktop/Study/REPORT/안효범 교수님/알고리즘/binay search$
```

사진 1: *binay-search* 배열 : 100000, 검색 숫자 : 1, time : 259810ns

2. 재귀적 호출 프로그램

```
pchero@MyNote: ~/Desktop/Study/REPORT/안호범 교수님/알고리즘/binay search
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)

원하시는 배열의 크기를 입력해 주세요.
배열의 크기 : 100000

찾고자 하시는 숫자를 입력해 주세요.
숫자 입력! : 1

잠시만 기다려 주세요....

현재 1번째의 루프를 수행중입니다.
현재 2번째의 루프를 수행중입니다.
현재 3번째의 루프를 수행중입니다.
현재 4번째의 루프를 수행중입니다.
현재 5번째의 루프를 수행중입니다.
현재 6번째의 루프를 수행중입니다.
현재 7번째의 루프를 수행중입니다.
현재 8번째의 루프를 수행중입니다.
현재 9번째의 루프를 수행중입니다.
현재 10번째의 루프를 수행중입니다.
현재 11번째의 루프를 수행중입니다.
현재 12번째의 루프를 수행중입니다.
현재 13번째의 루프를 수행중입니다.
현재 14번째의 루프를 수행중입니다.
현재 15번째의 루프를 수행중입니다.
현재 16번째의 루프를 수행중입니다.
현재 17번째의 루프를 수행중입니다.

찾았습니다!!!
배열의 인덱스의 위치는 1입니다.

재귀적 함수 실행 시간입니다.

실행시간 : 0s, 265956ns
pchero@MyNote:~/Desktop/Study/REPORT/안호범 교수님/알고리즘/binay search$
```

사진 2: *binay-recrsive* 배열 : 100000, 검색 숫자 : 1, time : 265956ns

5. 결론

1. 단순 호출 프로그램의 수행시간 : 259810ns
2. 재귀적 호출 프로그램 수행시간 : 26595ns
3. 재귀적 호출 프로그램 수행시간이 더 길다.