

# 자료구조응용 과제

\*LMS 공지사항에 응용과제 제출 시 파일명 정하는 법과 제출 시 주의 사항이 공지되어 있으니 읽어 주세요.  
\*실습실 PC에 코드가 남아있지 않도록 해 주세요. - 퇴실 시에 코드를 작성한 본인이 반드시 삭제해 주세요.

과제 #3 (만점: 10 점)

**단계 1)** 첨부된 파일 hw3\_in.txt에는 오름차순의 중복 없는 정수들이 저장되어 있다. hw3\_in.txt에 저장된 정수의 개수 (L\_size)를 알아낸 뒤, L\_size 개의 정수들을 저장할 수 있는 array L을 malloc으로 생성하라. 그 후, 파일 hw3\_in.txt로부터 정수를 차례대로 읽어 들여 array L에 저장한다.

- feof를 사용할 것.
- File에 저장된 모든 정수의 개수(L\_size)를 알아내기 위해서는 file을 open하여 모든 정수를 차례대로 읽으며 count해야 하고, 이때 file pointer는 file의 끝까지 가게 됨. File pointer를 file의 시작 위치로 다시 이동하기 위해서는 두 가지 방법이 가능함.
  - L\_size를 알아내기 위해 file open/close를 한번 실행하고, 그 뒤 in.txt의 정수들을 읽어들이기 위해 file open/close를 한번 더 실행하는 방법이 있음.
  - Fseek을 이용하여 file open 상태에서 pointer를 초기 위치로 이동할 수도 있음.

**단계 2)** 위에서 생성한 array L을 이용하여 sequential search, iterative binary search, recursive binary search의 time complexity를 비교하고자 한다. 0부터 freq-1까지의 모든 정수를 차례대로 array L에서 search하며, search한 결과를 모두 출력하면 출력량이 너무 많기 때문에 0부터 4까지 search한 결과만 출력하여 실행이 바르게 되었는지 확인한다.

function seqsearch(), iter\_binsearch(), rec\_binsearch()가 각각 sequential search, iterative binary search, recursive binary search를 수행하도록 작성하고 아래의 코드를 완성하라.

```
int main()
{
    // 학부, 학번, 이름 출력

    // 단계 1 작성 부분

    // 단계 2

    int freq = 100000;

    printf("sequential search:\n");
    start = clock();
    for (int i = 0; i < freq; i++) {
        int res = seqsearch(L, i, L_size);
        if (i < 5) printf("search %d result: %d\n", i, res);
    }
    end = clock();
    printf("computation time: %fms\n", 1000 * ((double)end - (double)start) / CLOCKS_PER_SEC);

    printf("iterative binary search:\n");
    // 위의 코드를 참고하여 작성

    printf("recursive binary search:\n");
    // 위의 코드를 참고하여 작성

    return 0;
}
```

실행 예제

```
Microsoft Visual Studio 디버그 콘솔
sequential search:
search 0 result: 0
search 1 result: 1
search 2 result: -1
search 3 result: 2
search 4 result: 3
computation time: 2047.000000ms

iterative binary search:
search 0 result: 0
search 1 result: 1
search 2 result: -1
search 3 result: 2
search 4 result: 3
computation time: 10.000000ms

recursive binary search:
search 0 result: 0
search 1 result: 1
search 2 result: -1
search 3 result: 2
search 4 result: 3
computation time: 37.000000ms
```

추가과제 #3 (만점 1. 응용 수업 시간 내에 위의 본과제를 검사 받고 제출한 후에만 추가과제의 채점 및 제출이 가능함. 추가과제는 응용수업 시간 내에 채점 받고 제출해야 하며 늦은 제출 불가함)

Recursion 을 이용하여 정수 N 의 binary representation 에 속한 ‘1’의 개수를 화면에 출력하라. N 의 binary representation 에 존재하는 ‘1’의 개수는, N 이 홀수인 경우에는  $\lfloor N/2 \rfloor$ 의 binary representation 에 있는 ‘1’의 개수 + 1 이다. 만약 N 이 짝수이면,  $\lfloor N/2 \rfloor$ 의 binary representation 에 있는 ‘1’의 개수와 같다. 정수 N (  $1 \leq N \leq 1024$ )은 scanf\_s 로 반복하여 입력 되도록 하고, N 이 -1 이면 수행을 종료한다.

<실행 예제>

Ex1	Ex2
컴퓨터학부 202420394 홍길동 (scanf_s) 16	컴퓨터학부 202420394 홍길동 (scanf_s) 1024
1	1
(scanf_s) 17	(scanf_s) 128
2	1
(scanf_s) 18	(scanf_s) 1
2	1
(scanf_s) -1	(scanf_s) -1