

자료구조응용

02. 공간복잡도, 스택과 힙, 파일 입출력 (11점)

2022.3.7

1. 다음과 같은 프로그램을 두 가지 버전으로 작성하시오. (3점)

(※ 지난 과제 마지막 문제 결과에 이진탐색을 추가하면 됩니다.)

▶▷실행순서

- ① 사용자로부터 난수생성 개수(n)를 입력받는다.
- ② 정수 난수를 n개 발생시켜 1차원 배열에 저장한다.
- ③ 1차원 배열에 대해 선택정렬(selection sort)을 수행한다.
- ④ 사용자로부터 임의의 정수를 입력받는다.
- ⑤ 입력받은 정수가 배열에 있는지 이진탐색(binary search)을 수행하여 그 결과를 출력한다.

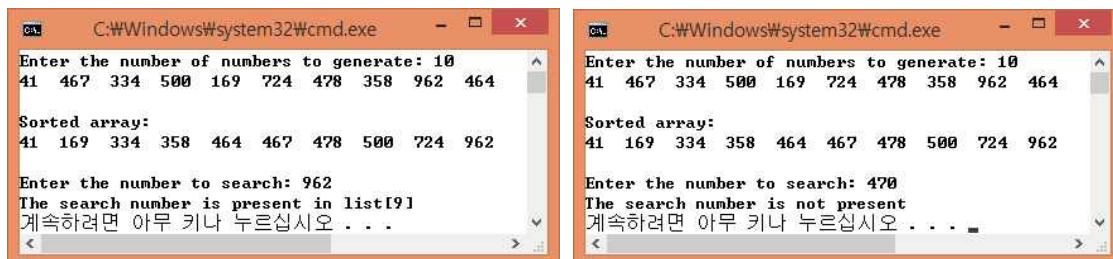
▶▷난수생성조건

난수생성 개수는 최대 100개, 난수 범위는 0~999, 난수 중복허용, SEED를 지정하지 않음

▶▷교재 참고프로그램

Program 1.3, 1.4, 1.6, 1.7, 1.8

▶▷실 행 예



(1) 함수 swap, compare와 반복문을 사용한 이진탐색을 구현한 버전

(2) 매크로 SWAP, COMPARE와 재귀호출을 사용한 이진탐색을 구현한 버전

2. 프로그램 P의 공간복잡도(space complexity) $S(P)$ 는 다음과 같이 정의된다. (5점)

$$S(P) = c + S_P(I)$$

c : 고정된 크기의 공간

$S_P(I)$: 프로그램 P의 인스턴스 I에 대한 가변크기 공간

(1) 다음 두 함수 sum, rsum에 대해 공간복잡도를 주어진 조건하에 비교해 보라.

조건 ① 두 함수에 대한 최초 호출이 sum(list, 10), rsum(list, 10)

② $S(P)$, c , $S_P(I)$ 를 각각 계산하기. 단, c 는 스택에 할당되는 고정크기만 계산

```
float sum(float list[], int n)
{
    float tempsum = 0;
    int i;
    for (i = 0; i < n; i++)
        tempsum += list[i];
    return tempsum;
}
```

Program 1.11: Iterative function for summing a list of numbers

```
float rsum(float list[], int n)
{
    if (n) return rsum(list, n-1) + list[n-1];
    return 0;
}
```

Program 1.12: Recursive function for summing a list of numbers

(2) 최초 호출이 rsum(list, 10)일 때, 가장 마지막에 rsum이 호출된 후 리턴하지 않은 상태에서 최초 호출을 포함하여 5번째 rsum의 지역창과 호출스택을 화면 캡처하라.

(3) list 배열 원소에 1 ~ 사용자로부터 입력된 값까지의 정수를 지정하여 sum과 rsum을 수행하여 그 결과를 출력하는 프로그램을 작성하라.

[실행예]

```
input >> 10
sum : 1에서 10까지의 정수 합은 ....입니다.
rsum: 1에서 10까지의 정수 합은 ....입니다.
```

[보고서형식]

(1) 공간복잡도 분석

	최초 호출: sum(list, 10)	최초 호출: rsum(list, 10)
스택 그림 ※ 최대로 공간을 할 당한 순간		※ 중복내용은 : 등으로 표현할 것
S(P)	S(sum) =	S(rsum) =
c		
S _p (I)	S _{sum} (n) =	S _{rsum} (n) =

(2) 디버거 화면캡처

(3) 실행예 화면캡처

3. 다음과 같은 파일 입출력 프로그램을 작성하라. 또한 ①~③ 함수 실행 후(즉, 반환 직후)의 스택 및 힙 상태에 대한 그림을 그려라. (3점)

[main 함수]

```
int main(void)
{
    int num, *pNum;

    FILE *fpIn = fopen("input.txt", "r");
    FILE *fpOut = fopen("output.txt", "w");

    fscanf(fpIn, "%d", &num); // 데이터 갯수
    pNum = alloc(num);        // num 크기 만큼 메모리 할당
    input(pNum, num, fpIn);   // 파일로부터 num개 데이터 입력 ①
    increase(pNum, num);      // 각 데이터를 1씩 증가 ②
    output(pNum, num, fpOut); // 증가된 데이터를 파일출력

    free(pNum); ③
    fclose(fpIn);
    fclose(fpOut);

    return 0;
}
```

[입출력 파일형식]

input.txt	output.txt
10	10
1 2 3 4 5 6 7 8 9 10	2 3 4 5 6 7 8 9 10 11

[보고서 형식]

	① input 실행 후	② increase 실행 후	③ free 실행 후
스택 및 힙 그림			
지 역 창 화면캡처			

[셀프체크]

Q1. 지역창에서 힙에 할당된 전체 값의 확인이 가능한가? 다른 방법?

Q2. 자동창은 어떤 정보를 보여주나?

■ 제출 형식

- 솔루션 이름 : DS 02

- 프로젝트 이름 : 1-1, 1-2, 2, 3

- 각 소스파일에 주석처리

“학번 이름”

“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

- 실행화면을 캡처한 보고서를 작성 후 pdf 파일로 변환하여 솔루션 폴더에 포함

- 솔루션 정리 메뉴를 수행 후 전체 솔루션을 “학번.zip”으로 압축하여 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!

- 1차 마감 (LMS 과제 마감일) : 수업일 자정

- 2차 마감 (LMS 과제 열람기한) : 수업 익일 자정(만점의 50%, 반올림)

- 1, 2차 마감 이외의 제출은 허용하지 않습니다. (이메일 제출 불가!)