

Tech University of Korea(TUK)

자료구조 과제 노트

202#-0#학기

담당교수	박정민 교수님
학번	•
이름	• 신##

과제수행 요청서

과제기간

■ (과제기간) 과제는 교재의 각 장이 끝난 일주일 후, 24시까지 제출

- (제출방법) e-class의 ▲과제게시판에 업로드
- (제출파일명) [학번이름].hwp 예시) 01[2017131023박정민].hwp
- (지각제출불가) 과제를 제출하는 기간에 반드시 제출, ▲지각제출불가
- (개인제출) 과제제출은 ▲개인적으로 수행

과제내용

■ (과제내용) 과제의 내용은 총 4가지 1)요점정리, 2)역공부, 3)순공부, 4)자기성찰

- (요점정리) 수업시간의 내용을 ▲재정리 ▲ 개별적으로 공부한 것이 있다면 추가 요점정리
- (역공부) 코드를 분석한 내용정리 ▲손으로 분석한 그림, ▲디버깅 SW를 이용한 분석그림
- (순공부) 역공학을 통해 분석한 소스코드를 근거로 ▲코드를 수정/개선시켜보기, ▲주석 상세히 달기
- (자기성찰) ▲수업을 통해서 배운 것, ▲디버깅을 통해 집중적으로 공부한 것

결과물활용

■ (결과물) 결과물은 1)수시고사, 2)과제평가, 3) 개인 정리를 위한 지침

- (수시고사를 위한 참조) 과제들은 수시고사와 연계 ▲오픈북 시에 참조 문서로 활용
- (과제평가) 각 장별로 자료구조 수업이 종강될 때까지 작성하여 매주 평가
- (개인정리를 위한 지침) 과제내용 스스로 잘 정리하기 위한 좋은 가이드라인으로 활용

자료구조 과제 목차

1. 제1장 자료구조와 알고리즘

1-1 1장 자료구조와 알고리즘 요점정리

1-2 소스코드 디버깅 분석(역공부)

1-3 소스코드 수정/개선 & 주석(순공부)

2. 자기성찰

2-1 평가내용 및 느낀 점 (총 50점)

1. 제1장 자료구조와 알고리즘

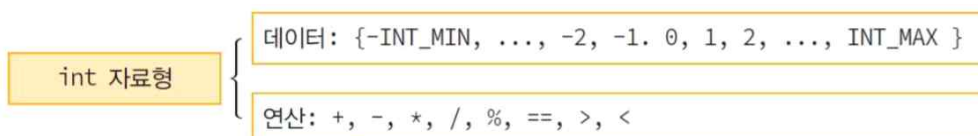
1-1 1장 자료구조와 알고리즘 요점정리

■ 자료구조와 알고리즘

- 프로그램은 1) 자료구조와 2) 알고리즘의 조합으로 이뤄진다.
- 1) 자료구조의 종류는 스택, 큐, 리스트, 사전, 그래프, 트리가 있다.
ex) ○○과목의 학생들 점수들의 배열
- 2) 알고리즘은 컴퓨터로 문제를 풀기 위한 단계적인 절차이다. (특정 함수의 진행 절차)
ex) ○○과목의 학생들 점수들의 배열의 값 중에서 최종적으로 우수한 점수를 추려냄.
- 알고리즘의 조건에는 입력(0개 이상 존재), 출력(1개 이상 존재), 처리가 있다.
ex) `int find_max_score(int n) {...}` → 입력은 정수 `n`, 출력은 `int`형 정수이며 처리는 우수 점수이다.
- 알고리즘 조건은 이 외에도 모호하지 않고 명백한 지, 유한한 지, 유효성이 있다.
- 알고리즘의 기술 방법은 총 4가지가 있다.
 - 1) 자연어: 인간이 읽기 쉽지만 의미 전달이 모호해지는 단점이 존재한다.
 - 2) 순서도(흐름도): 간단하면 직관적이기 때문에 이해가 쉽지만 복잡하면 기술이 어렵다.
 - 3) 유사 코드(의사 코드): 알고리즘 핵심적인 내용에만 집중할 수 있으며 프로그래밍 언어와 유사하다.
 - 4) 프로그래밍 언어: 알고리즘의 가장 정확한 기술이지만 핵심적인 내용에 대한 이해를 방해할 수 있다.
- 알고리즘의 성능분석에는 두 가지가 존재한다.
 - ① 직접 접근: `time.h` 을 이용, 알고리즘 시작과 끝의 시간 차이를 토대로 수행 시간을 측정한다.
 - ② 간접 접근: 빅오 표기법(연산의 횟수를 대략적으로 표기한 것, 영향이 작은 수는 상대적으로 무시한다.)
- 알고리즘의 성능분석이 중요한 이유는 프로그램의 효율성 때문이다.
- 알고리즘의 최선, 평균, 최악의 경우는 입력 자료 집합에 따른 수행 시간에 따라 다르다.
 - 최선의 경우: 수행 시간이 가장 빠르다.
 - 평균의 경우: 수행 시간이 평균적이다.
 - 최악의 경우: 수행 시간이 가장 늦다.
- 최선은 의미가 없는 경우가 많다. 따라서 최악의 경우를 널리 사용하며 중요한 의미를 가진다.
- 프로그램의 안전성을 측정하기 위해서는 최악의 경우를 토대로 한다.
 - 왜냐하면 최악의 경우에서 안전성 검사가 입증되면 최선, 평균의 경우에서도 문제가 없기 때문이다.

■ 자료형과 추상 데이터

- 자료형이란 데이터의 집합과 연산의 집합이다.



- 추상 데이터는 데이터 타입을 추상적으로 정의한 것이다.

- 추상 데이터는 추상화를 하기 위한 목적이다.
- 추상화는 사용자가 프로그램을 실행하는데는 문제가 없지만 세부 구현 사항은 공개되지 않는 것이다.

ex) TV의 볼륨 조정 키, 채널 조정 키는 사용자의 편리만 해결해주며 TV 내부의 데이터에는 접근을 할 수 없으며 어떻게 구현이 되는지 몰라도 이용할 수 있다.

1-2 소스코드 디버깅 분석(역공부)

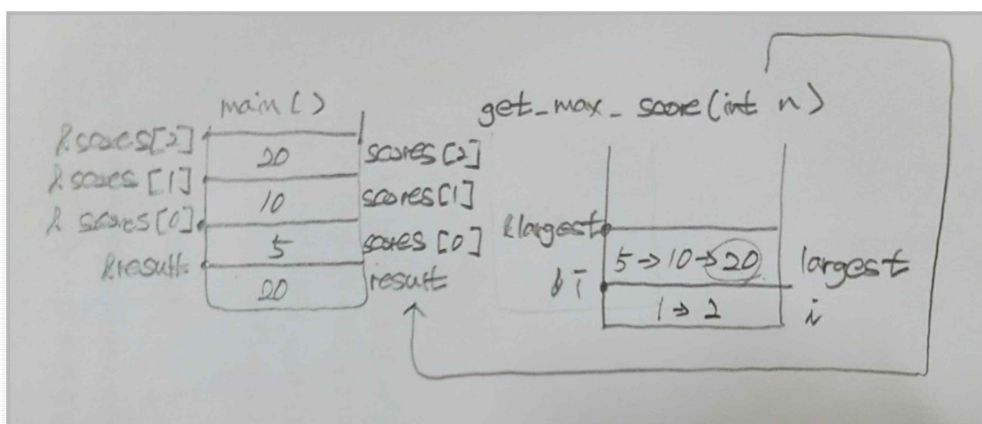
■ '순' 분석내용

- calc_scores.c

```

1  #include <stdio.h>
2  #define MAX_ELEMENTS 3
3  int scores[MAX_ELEMENTS]; // 자료구조, 전역배열
4
5  int get_max_score(int n); // 입력: n, 처리: get_max_score, 출력: int
6  // 함수의 선언, 함수의 리스트
7
8  int main(void)
9  {
10     int result;
11     scores[0] = 5;
12     scores[1] = 10;
13     scores[2] = 20;
14     result = get_max_score(MAX_ELEMENTS);
15     printf("결과:%d\n", result);
16     return 0;
17 }
18
19 int get_max_score(int n) // 학생의 숫자는 n
20 {
21     int i, largest;
22     largest = scores[0]; // 알고리즘
23     for (i = 1; i < n; i++)
24     {
25         if (scores[i] > largest)
26         {
27             largest = scores[i];
28         }
29     }
30     return largest;
31 }

```



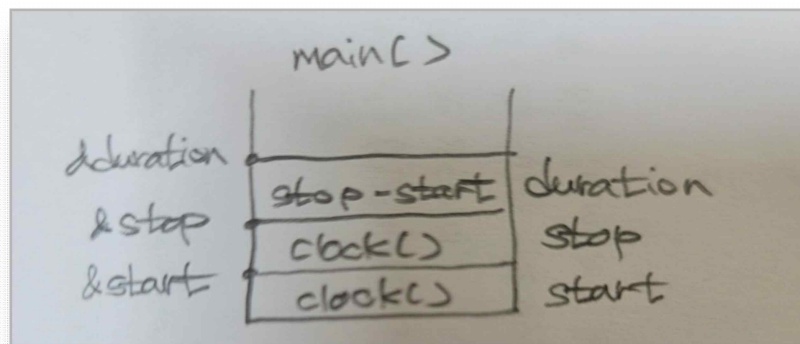
- calc_time.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 1000000

int main(void)
{
    clock_t start, stop;
    double duration;
    start = clock();    // 측정 시작

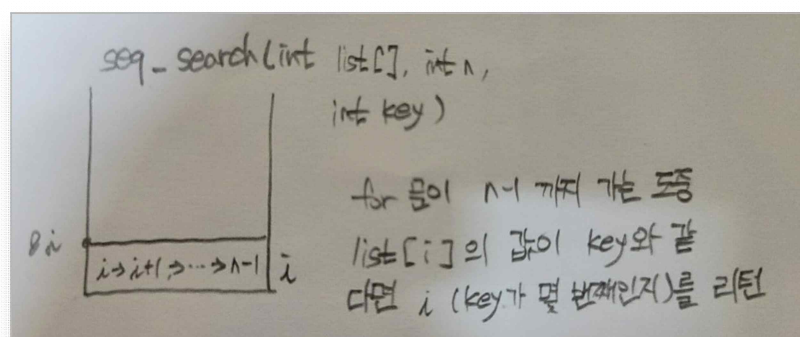
    for (int i = 0; i < MAX; i++)    // 의미 없는 반복 루프
        ;

    stop = clock();    // 측정 종료
    duration = (double)(stop - start) / CLOCKS_PER_SEC;
    printf("수행시간은 %f초입니다.\n", duration);
    return 0;
}
```



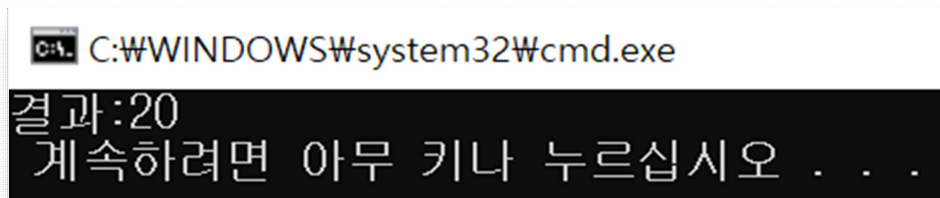
- seq_search.c

```
int seq_search(int* list, int n, int key)
{
    int i;
    for (i = 0; i < n; i++)
    {
        if (list[i] == key)
            return i;
    }
    /* 탐색에 성공하면 키 값의 인덱스 반환 */
    return -1;    /* 탐색에 실패하면 -1 반환 */
}
```

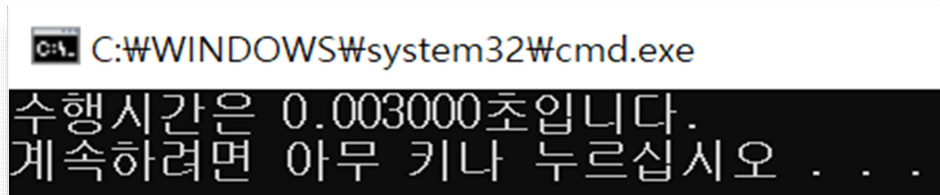


■ 디버깅SW를 이용한 분석내용

- calc_scores.c
- 1) int형 변수 result, scores, scores[0] 가 생성된다.
- 2) scores[0] 에 정수 5가 대입되며 scores[1]이 생성되었다.
- 3) scores[1] 에 정수 10가 대입되며 scores[2]이 생성되었다.
- 4) scores[2] 에 정수 20가 대입되었다.
- 5) MAX_ELEMENTS(최대 학생 수)가 3으로 정의되었기 때문에 get_max_score 함수 중 입력 n에 3이 대입된다.
- 6) int형 변수 largest가 생성되고 이때 scores[0]의 값 5가 대입되었다.
- 7) for문 i = 1부터 n까지 반복된다.
- 8) for문 안에 if 문은 for문이 돌아가면서 scores 배열에 저장된 값을 largest 값과 비교했을 때 scores[i] 의 값이 largest 값보다 크면 이 때의 scores[i]의 값이 largest에 저장된다.
- 9) 최종적으로 largest 값을 리턴 받아서 main 함수의 result 변수에 저장된다.
- 10) printf 함수를 이용하여 result 값(= scores 배열의 값들 중 가장 큰 값) 20을 출력한다.



- calc_time.c
- 1) time.h(헤더 파일)을 include 하였기 때문에 clock_t형 start, stop 변수가 생성되었다.
- 2) double 형 duration 변수를 생성하였다.
- 3) 알고리즘의 시작 부분의 시간을 start 변수에 저장하였다.
- 4) 의미 없는 반복 루프 for문(알고리즘)을 i = 0부터 10000까지 돌린다.
- 5) clock()을 이용하여 알고리즘의 종료 부분의 시간을 stop 변수에 저장하였다.
- 6) duration 변수에 알고리즘의 실행시간을 저장한다.
실행 시간 계산 코드 = (double)(stop - start) / CLOCKS_PER_SEC
- 7) printf 함수를 이용하여 duration 의 수행 시간(double 형이므로 %lf)을 화면에 출력한다.



1-3 소스코드 수정/개선 & 주석(순공부)

■ calc_scores.c (수정/개선 & 주석 및 설명) - 수정

<pre> 1 #include <stdio.h> 2 #define MAX_ELEMENTS 3 3 int scores[MAX_ELEMENTS]; // 자료구조, 전역배열 4 5 int get_max_score(int n); // 입력: n, 처리: get_max_score, 출력: int 6 // 함수의 선언, 함수의 리스트 7 8 int main(void) 9 { 10 int result; 11 scores[0] = 5; 12 scores[1] = 10; 13 scores[2] = 20; 14 result = get_max_score(MAX_ELEMENTS); 15 printf("결과: %d\n", result); 16 return 0; 17 } 18 19 int get_max_score(int n) // 학생의 숫자는 n 20 { 21 int i, largest; 22 largest = scores[0]; // 알고리즘 23 for (i = 1; i < n; i++) 24 { 25 if (scores[i] > largest) 26 { 27 largest = scores[i]; 28 } 29 } 30 return largest; 31 } </pre>	<pre> #include <stdio.h> #define MAX_ELEMENTS 3 int scores[MAX_ELEMENTS]; // 자료구조, 전역배열 int get_max_score(int n); // 입력: n, 처리: get_max_score, 출력: int // 함수의 선언, 함수의 리스트 int main(void) { int result; for (int i = 0; i < MAX_ELEMENTS; i++) { scanf("%d", &scores[i]); } // 사용자로부터 scores[n]의 값을 직접 받음 (개선 작업) // ... result = get_max_score(MAX_ELEMENTS); // int get_max_score() 함수가 완료되면 largest를 이 함수가 리턴받음 printf("결과: %d\n", result); return 0; } </pre>
기존 코드	개선된 코드

- 수정 함수: main()
- 수정 부분: scores 배열의 값을 각각 지정해 주지 않고 for 문을 돌려서 사용자로부터 scores 배열의 값을 지정한다.
- 추가 수정할 수 있는 부분: result = get_max_score(MAX_ELEMENTS) 코드 전후로 알고리즘 실행 시간을 측정하여 프로그램의 효율성 증진 방향을 고민해 볼 수 있다.

■ calc_scores.c (수정/개선 & 주석 및 설명) - 개선

- 개선 방향: calc_scores.c, calc_time.c, seq_search.c을 통합하여 N명의 학생 중 가장 높은 점수를 받은 학생을 출력하고 그 학생의 학번과 알고리즘의 실행 시간을 구한다.
(이때, 학번의 순서는 0번째부터 N-1(학생 수 - 1)번째 까지이다.)

- 개선 후 구현 순서
 1. scanf()을 이용하여 학생 수를 입력받는다.
 2. for()과 scanf()을 적절히 조합하여 1번째부터 N번째 학생의 점수를 입력받는다.
 3. 가장 높은 점수를 색출하고 그 점수를 받은 학생의 학번을 출력한다.
 4. 알고리즘의 실행 시간을 출력한다.

- main()

→ main()만 개선하였기 때문에 get_max_score(), seq_search()는 생략하였습니다.

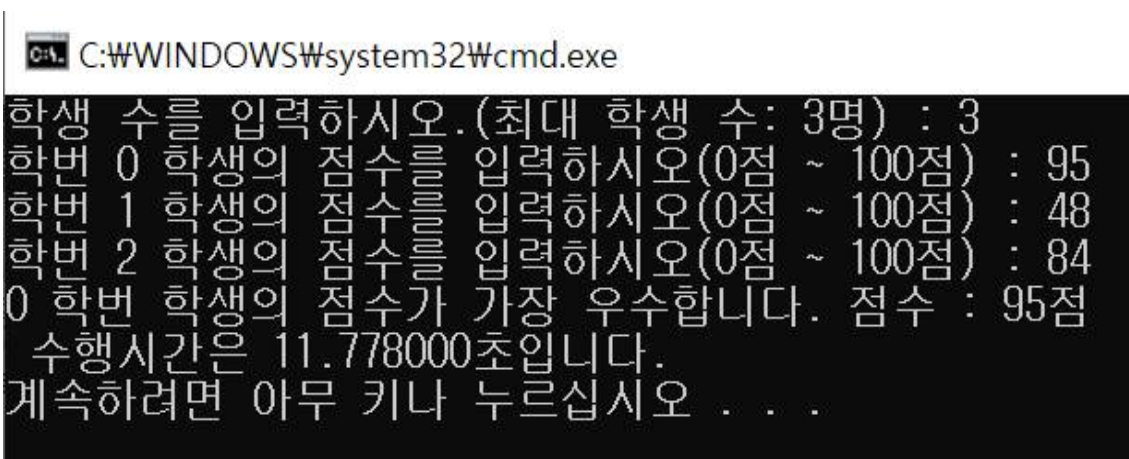
```
int main(void)
{
    int result;
    int stu, num;
    clock_t start, stop;
    double duration;
    start = clock();    // 측정 시작

    printf("학생 수를 입력하시오.(최대 학생 수: 3명) : ");
    scanf("%d", &stu);

    for (int i = 0; i < stu; i++) {
        printf("학번 %d 학생의 점수를 입력하시오(0점 ~ 100점) : ", i);
        scanf("%d", &scores[i]);
    } // 사용자로부터 scores[n]의 값을 직접 받음 (개선 작업)
    result = get_max_score(MAX_ELEMENTS); // int get_max_score() 함수가 완료되면 largest 를 이 함수가 리턴받음
    num = seq_search(scores, stu, result);
    printf("%d 학번 학생의 점수가 가장 우수합니다. 점수 : %d점\n", num, result);
    return 0;
}
```

- 1) 사용자로부터 학생 수를 입력받는다. (최대 학생 수는 MAX_ELEMENTS로 정의되어있다.)
- 2) 학번 0의 학생부터 N-1 학번의 학생까지의 점수를 입력받는다.
- 3) get_max_score 함수를 실행하여 가장 높은 점수를 result 변수에 대입한다.
- 4) seq_search 함수를 실행하여 가장 높은 학생의 점수의 학번을 찾는다.
- 5) printf()를 이용하여 원하는 정보를 화면에 출력한다.
- 6) 알고리즘의 실행 시간이 출력된다.

- 실행 결과



```
C:\WINDOWS\system32\cmd.exe
학생 수를 입력하시오.(최대 학생 수: 3명) : 3
학번 0 학생의 점수를 입력하시오(0점 ~ 100점) : 95
학번 1 학생의 점수를 입력하시오(0점 ~ 100점) : 48
학번 2 학생의 점수를 입력하시오(0점 ~ 100점) : 84
0 학번 학생의 점수가 가장 우수합니다. 점수 : 95점
수행시간은 11.778000초입니다.
계속하려면 아무 키나 누르십시오 . . .
```

- 느낀 점: 알고리즘의 효율성을 측정하는 방법 중 하나인 직접 측정을 이용하였다. 알고리즘의 시작과 끝 시간을 토대로 시간을 측정하였더니 사용자로부터 값을 입력받는 경우 입력받는 시간도 포함된다.

2. 자기성찰

2-1 평가내용 및 느낀 점 (총 50점)

평가 내용 (30점 만점)	3점	2점	1점
1. 온라인강의는 주어진 기한 내 듣기 완료를 하였나?	●		
2. 이번 강의 수업은 이해도가 높았나?	●		
3. 역공부, 주어진 소스코드 분석을 위해 손으로 그림을 그려보았나?	●		
4. 역공부, 주어진 소스코드 분석을 위해 디버깅 SW로 분석해보았나?	●		
5. 역공부, 소스코드를 구현한 개발자의 노력을 생각해보았나?		●	
6. 순공부, 3번과 4번을 수행하면서 코드를 개선시킬 노력은 하였나?	●		
7. 순공부, 개선코드 주석은 꼼꼼히 작성하였나?	●		
8. 과제 '양식(hwp)'을 준수하여 작성하였나?	●		
9. 오타 없이 표의 양식, 그림의 넓이&높이 등 정돈된 보고서인가?	●		
10. 시간 마감에 급급한 대충 작성한 보고서가 아닌, 열심히 공부한 정직한 보고서인가?	●		
총 점	29 점		

‘객관식 30점, 주관식 20점, 총 50점을 스스로 자기성찰’ 하고, 정직한 보고서를 위해 최선을 다하자!

노력지수 (10점 만점)	<ul style="list-style-type: none"> • 8점 <ul style="list-style-type: none"> - 이전과는 다르게 이번에는 온라인 강의를 듣고 보는 것에 그치지 않고 옆 화면에 비주얼 스튜디오를 실행하여 교수님 진도에 맞춰서 같이 실행해보았다. 그랬더니 집 중도 잘되고 교수님께서 하신 부분과 내가 했을 때 헛갈리거나 어긋날 때는 일시정지하여 무엇이 잘못되었는지 이해하려고 노력했다. 아직까지는 교수님 진도에 맞춰가기 수월했다. 강의 및 과제를 열심히 수행했다고 생각한 점, 코드 개선에서 코드 줄이기에 초점을 둔 점을 반영하여 노력지수를 매겼습니다.
느낀 점 (10점 만점)	<ul style="list-style-type: none"> • 8점 <ul style="list-style-type: none"> - 자료구조와 알고리즘의 기본적인 이론과 예시를 배웠습니다. 알고리즘의 수행시간이 효율성을 증진하는데 중요하다는 것을 깨달았고 주어진 코드를 적절히 조합하여 기초적이지만 완성도 있는 프로그램을 만든 것 같아서 뿌듯합니다. 아직은 1주차이기 때문에 기본적인 내용만 다룬다는 점을 인지하여 게을리하게 과제 수행 및 강의를 들어서는 안되겠다고 생각했습니다. 자료구조는 컴퓨터공학에서 매우 중요한 부분이라는 교수님들의 말씀을 토대로 앞으로의 수업도 항상 성실하게 들어야겠다는 느낌을 받았습니다.

- ‘자기성찰’ 부분은 과제 채점 시, 많은 부분 참고가 됩니다. -