

자료구조응용

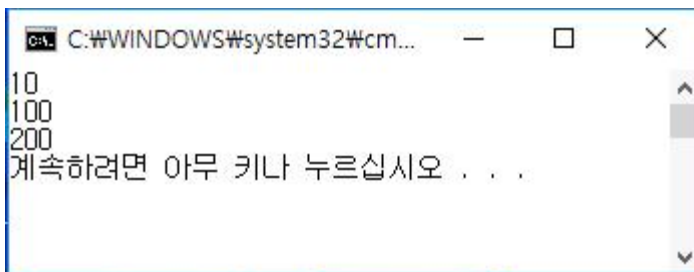
chapter 01-2

1. 다음 프로그램의 func1, func2, func3을 정의한 후 실행되도록 구현하라.

```
int main(void)
{
    int *pNum;
    pNum = func1(); // int 크기만큼 동적할당 받아 10을 지정후 정수형 포인터 리턴
    printf("%d\n", *pNum);

    func2(pNum); // pNum이 가리키는 것을 100으로 수정
    printf("%d\n", *pNum);

    func3(&pNum); // pNum이 가리키는 것을 200으로 수정
    printf("%d\n", *pNum);
    free(pNum);
    return 0;
}
```



2. 다음 Program 1.24를 사용하여 선택정렬(Selection Sort)에 대한 성능측정을 하라. 그리고 프로그램의 실행결과를 이용하여 그래프를 작성하라.

cf) 파일 처리 : out.txt 파일에 결과 값을 출력

- file open

fopen을 스트림을 직접 대입, fopen_s는 전달 인자로 FILE 포인터의 주소를 받아 스트림을 생성

```
FILE *fp = fopen("out.txt", "w");
```

```
FILE *fp;
```

```
fopen_s(&fp, "out.txt", "w");
```

- formatted print in file

```
fprintf(fp, "%6d    %f\n", n, duration);
```

- file close

```
fclose(fp);
```

[프로그램 설명]

- "selectionSort.h"는 Program 1.4의 SWAP 정의문, sort 함수를 참고하여 작성하기
- 화면출력 및 파일출력(out.txt)을 동시에 하도록 소스를 수정하기
- n을 2000까지 증가시켜 데이터를 생성하도록 소스를 수정하기

```
#include <stdio.h>
#include <time.h>
#include "selectionSort.h"
#define MAX_SIZE 1001
void main(void)
{
    int i, n, step = 10;
    int a[MAX_SIZE];
    double duration;
    clock_t start;

    /* times for n = 0, 10, ..., 100, 200, ..., 1000 */
    printf("    n    time\n");
    for (n = 0; n <= 1000; n += step)
    { /* get time for size n */

        /* initialize with worst-case data */
        for (i = 0; i < n; i++)
            a[i] = n - i;

        start = clock( );
        sort(a, n);
        duration = ((double) (clock() - start))
                    / CLOCKS_PER_SEC;
        printf("%6d    %f\n", n, duration);
        if (n == 100) step = 100;
    }
}
```

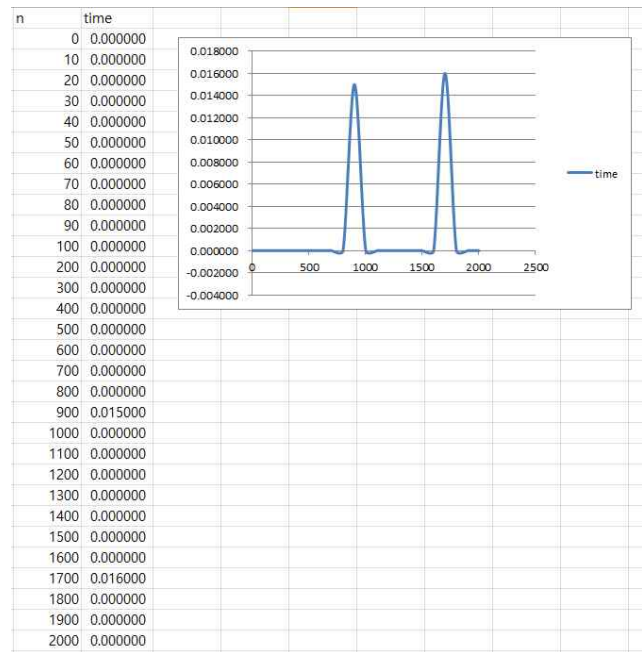
Program 1.24: First timing program for selection sort

[실행결과(out.txt)에 대한 그래프작성 예]

out - 메모장

파일(F) 편집(E) 서식(O)
보기(V) 도움말(H)

n	time
0	0.000000
10	0.000000
20	0.000000
30	0.000000
40	0.000000
50	0.000000
60	0.000000
70	0.000000
80	0.000000
90	0.000000
100	0.000000
200	0.000000
300	0.000000
400	0.000000
500	0.000000
600	0.000000
700	0.000000
800	0.000000
900	0.015000
1000	0.000000
1100	0.000000
1200	0.000000
1300	0.000000
1400	0.000000
1500	0.000000
1600	0.000000
1700	0.016000
1800	0.000000
1900	0.000000
2000	0.000000



3. 다음 Program 1.25를 사용하여 선택정렬(Selection Sort)에 대한 성능측정을 하라. 그리고 프로그램의 실행결과를 이용하여 그래프를 작성하라. 또한, 문제 2의 결과와 같이 하나의 그래프로 작성해서 비교해 보라.

[프로그램 설명]

- 위 1번과 같음

```
#include <stdio.h>
#include <time.h>
#include "selectionSort.h"
#define MAX_SIZE 1001
void main(void)
{
    int i, n, step = 10;
    int a[MAX_SIZE];
    double duration;

    /* times for n = 0, 10, ..., 100, 200, ..., 1000 */
    printf("    n    repetitions    time\n");
    for (n = 0; n <= 1000; n += step)
    {
        /* get time for size n */
        long repetitions = 0;
        clock_t start = clock();
        do
        {
            repetitions++;

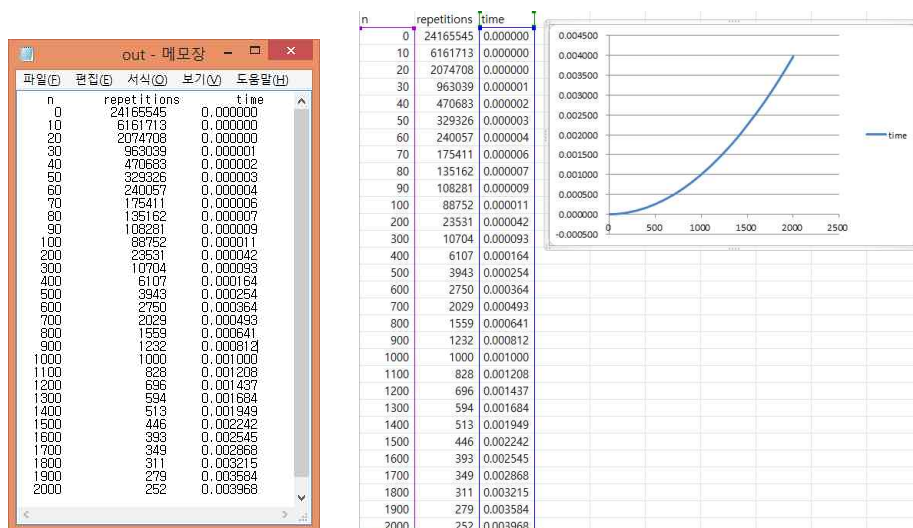
            /* initialize with worst-case data */
            for (i = 0; i < n; i++)
                a[i] = n - i;

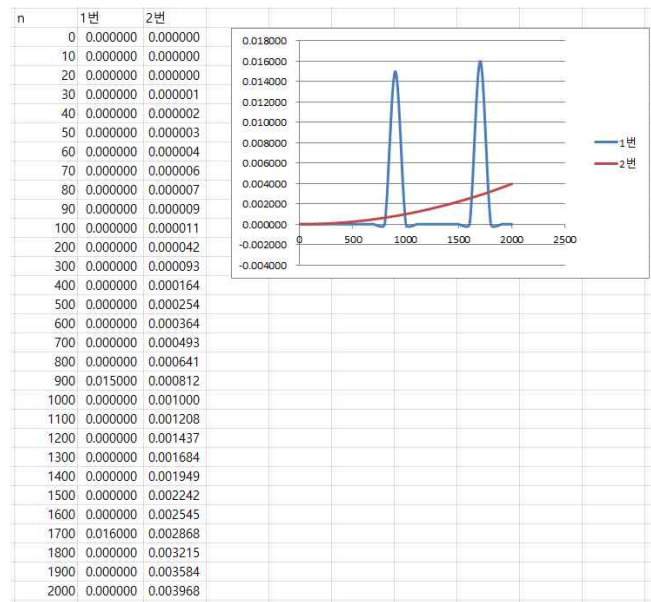
            sort(a, n);
        } while (clock() - start < 1000);
        /* repeat until enough time has elapsed */

        duration = ((double) (clock() - start))
                    / CLOCKS_PER_SEC;
        duration /= repetitions;
        printf("%6d %9d %f\n", n, repetitions, duration);
        if (n == 100) step = 100;
    }
}
```

Program 1.25: More accurate timing program for selection sort

[실행결과(out.txt)에 대한 그래프작성 예]



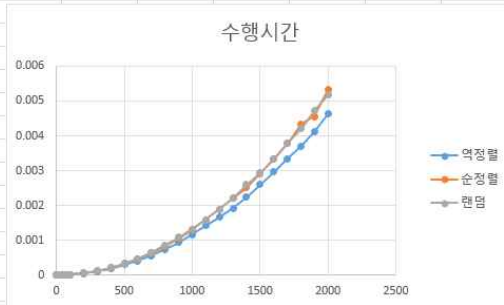


4. 3번 소스에 대해 데이터 생성부분을 일부 수정하여 다음 세 가지 경우에 대해 성능측정을 하여 비교하라. 그리고 selection sort 알고리즘의 경우 worst-case, best-case, average-case data의 구분이 있는지 기술하라.

- (1) 입력데이터가 내림차순 정렬이 이미 되어 있는 경우(3번 결과 사용)
- (2) 입력데이터가 오름차순 정렬이 이미 되어 있는 경우 [프로젝트명: 4-2]
- (3) 입력데이터를 난수생성으로 만들어 사용하는 경우 [프로젝트명:4-3]

[실행결과에 대한 그래프작성 예]

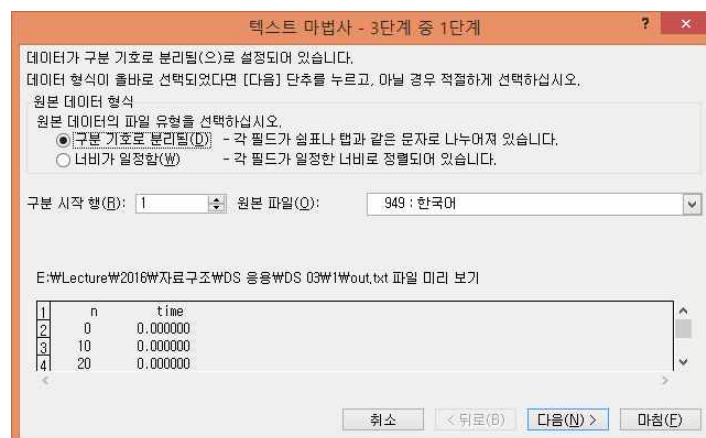
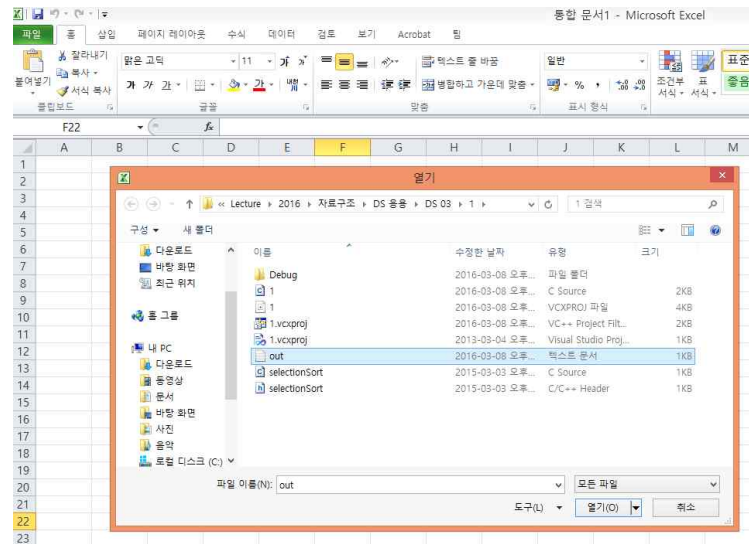
A	B	C	D	E	F	G	H	I	J	K	L
	n size	역정렬	순정렬	랜덤							
	0	0	0	0							
	10	0	0	0							
	20	0.000001	0.000001	0.000001							
	30	0.000001	0.000001	0.000002							
	40	0.000002	0.000002	0.000003							
	50	0.000003	0.000004	0.000005							
	60	0.000004	0.000005	0.000007							
	70	0.000006	0.000007	0.000008							
	80	0.000008	0.000009	0.00001							
	90	0.00001	0.000011	0.000013							
	100	0.000012	0.000013	0.000015							
	200	0.000047	0.000053	0.000056							
	300	0.000106	0.00012	0.000119							
	400	0.000183	0.000208	0.000208							
	500	0.000294	0.000322	0.000326							
	600	0.000407	0.000463	0.000471							
	700	0.000556	0.000637	0.000642							
	800	0.000743	0.000832	0.000858							
	900	0.000939	0.001052	0.001079							
	1000	0.001157	0.001302	0.001318							
	1100	0.001414	0.001579	0.001582							
	1200	0.001655	0.001883	0.001885							
	1300	0.001921	0.0022	0.00221							
	1400	0.002229	0.002509	0.002597							
	1500	0.002593	0.002915	0.00293							
	1600	0.002965	0.003344	0.003333							
	1700	0.003343	0.003785	0.003792							
	1800	0.003711	0.004333	0.00421							
	1900	0.004107	0.00455	0.004726							
	2000	0.004622	0.00533	0.00517							



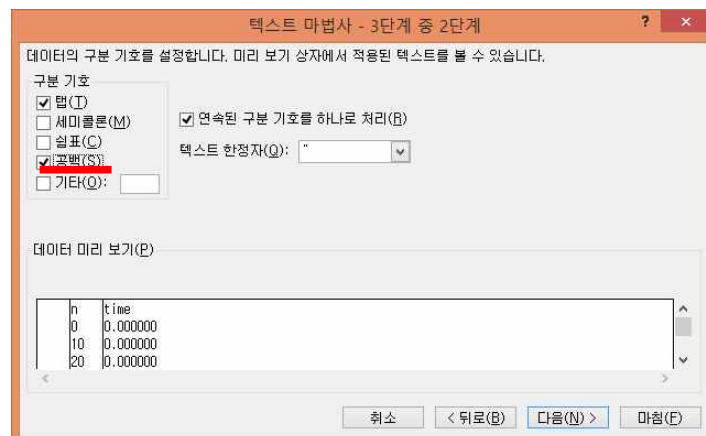
■ 참고 : 실행결과 파일로부터 표와 그래프 만들기

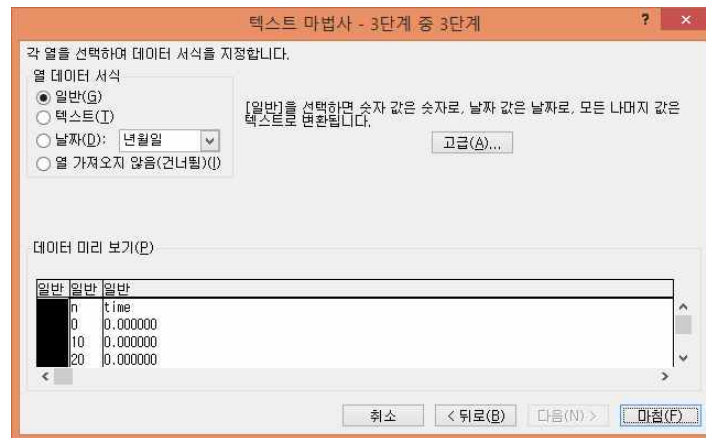
① 엑셀의 파일>열기로 출력파일(out.txt)을 연다.

(주의: 엑셀 실행 후 출력파일(out.txt)을 마우스 drag&drop 으로 오픈하면 안 됨)



(※ 경우에 따라 “너비가 일정함”으로 진행되는 경우도 있음)

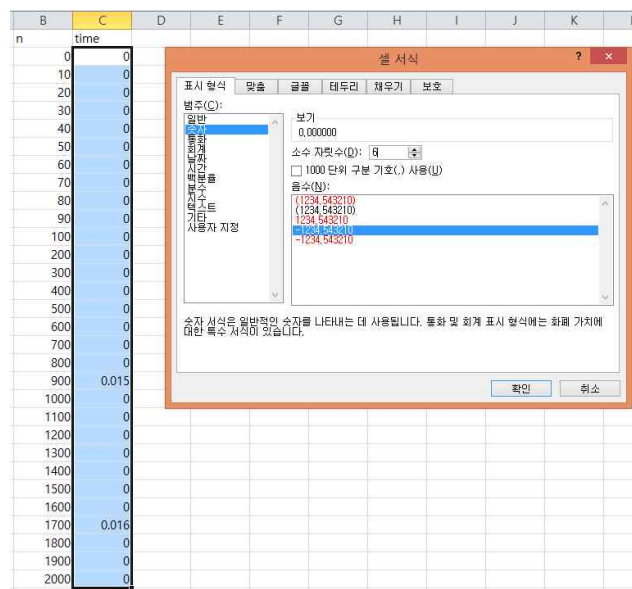




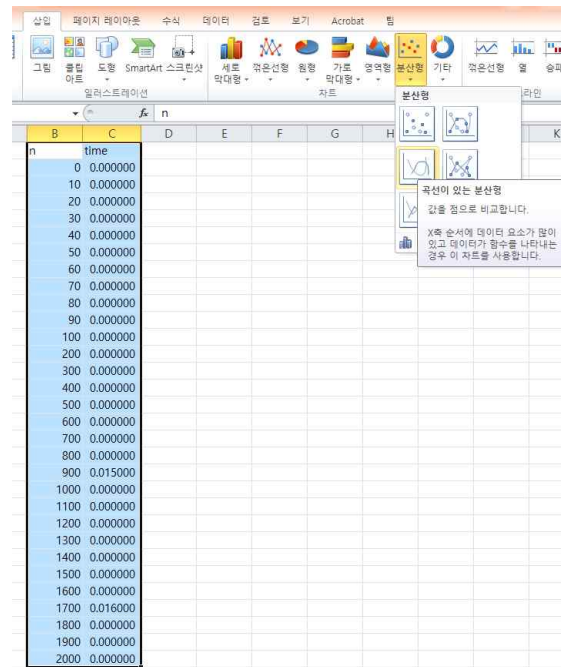
	A	B	C
1		n	time
2		0	0
3		10	0
4		20	0
5		30	0
6		40	0
7		50	0
8		60	0
9		70	0
10		80	0
11		90	0
12		100	0
13		200	0
14		300	0
15		400	0
16		500	0
17		600	0
18		700	0
19		800	0
20		900	0.015
21		1000	0
22		1100	0
23		1200	0
24		1300	0
25		1400	0
26		1500	0
27		1600	0
28		1700	0.016
29		1800	0
30		1900	0
31		2000	0

② 파일 > 다른이름으로 저장을 실행 후, Excel 통합문서 형식으로 저장한다.

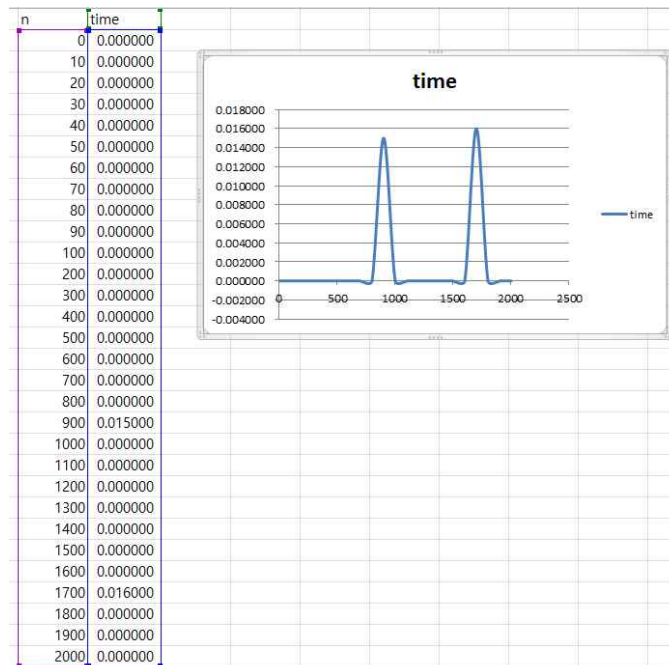
③ time 데이터의 셀서식을 “숫자, 소수 자리수 6”으로 변경한다.



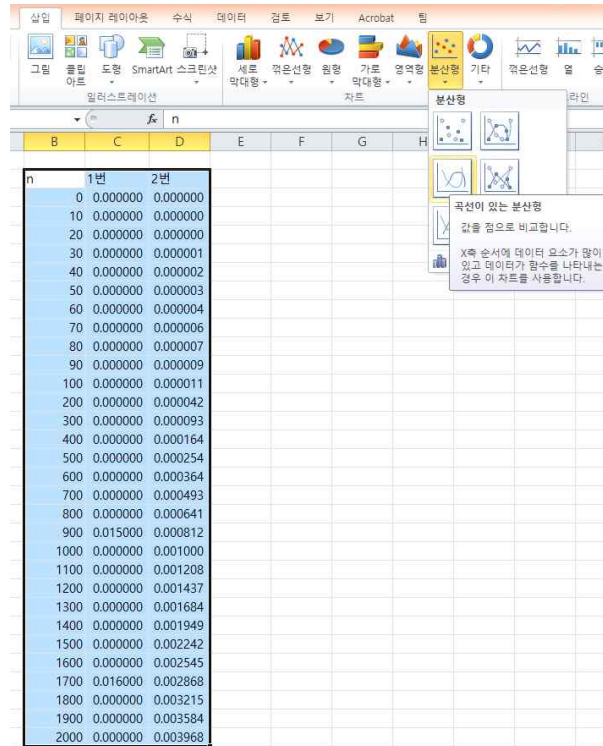
④ 아래와 같이 블록을 지정한 후 “삽입 > 분산형 > 곡선이 있는 분산형” 아이콘을 선택하기



⑤ 그래프 생성 결과



※ 그래프를 겹쳐서 그리기



■ 제출 형식

- 공학인증 시스템(ABEEK)에 과제를 올릴 때 제목:
- 1차 제출: 학번_이름_DS_02(1), 2차 제출: 학번_이름_DS_02(2)
- 솔루션 이름 : DS_02
- 프로젝트 이름 : 1, 2, 3, 4
- 실행화면을 캡처하여 한글파일에 추가 후 솔루션 폴더에 포함.
- 한글 파일명 : 학번_이름.hwp
- 솔루션 폴더를 압축하여 제출할 것.
- 솔루션 압축 파일 명:
 - 1차 제출: 학번_이름_DS_02(1).zip, 2차 제출: 학번_이름_DS_02(2).zip
- 제출은 2회 걸쳐 가능(수정 시간 기준으로 처리)