

REPORT

Hanseo University



선택 정렬 함수의 시간 측정 프로그램

과목: 자료구조

학과: 항공컴퓨터전공

학년: 2학년

학번: 202200654

이름: 김효준

수강 요일: 월요일

제출일: 2023-04-10

1.

```
1 #include<stdio.h>
2 #include<time.h>
3 #include "selectionSort.h"
4 #define MAX_SIZE 1001
5
6 void main(void){
7     int i, n, step = 10;
8     /*
9     n: a배열의 갯수
10    step: n을 증가시킬 갯수
11    */
12    int a[MAX_SIZE];
13    double duration;
14    clock_t start; //시작한 클럭 시간
15
16    printf("    n    time\n");
17    for(n=0;n<=1000;n+=step){
18        for(i=0;i<n;i++)
19            a[i]=n-i;
20        start = clock();
21        sort(a, n);
22        duration = ((double) (clock() - start)) / CLOCKS_PER_SEC; //걸린 시간
23        printf("m    %f\n", n, duration);
24        if(n==100) step = 100; //n이 100이 되면 100씩 증가시키기 시작.
25    }
26}
```

n	time
0	0.000001
10	0.000000
20	0.000001
30	0.000002
40	0.000002
50	0.000003
60	0.000004
70	0.000005
80	0.000007
90	0.002602
100	0.000010
200	0.000065
300	0.000083
400	0.000214
500	0.000285
600	0.000307
700	0.000459
800	0.000561
900	0.000719
1000	0.000900

2.

```
1 #include <stdio.h>
2 #include <time.h>
3 #include "selectionSort.h"
4 #define MAX_SIZE 1001
5
6 void main() {
7     int i, n, step=10;
8     /*
9     n: 배열의 갯수
10    step: n을 증가시킬 갯수
11    */
12    int a[MAX_SIZE];
13    double duration; //측정한 시간
14
15    /*times for n = 0, 10, ... 100, 200, ..., 1000*/
16    printf("    n    repetitions    time\n");
17    for(n=0;n<=1000;n+=step){
18        /*get time for size n*/
19        long repetitions = 0; //반복한 수
20        clock_t start = clock();
21        do{
22            repetitions++;
23
24            /*initialize with worst-case data*/
25            for(i=0;i<n;i++)
26                a[i]=n-i;
27            sort(a, n);
28        } while(clock()-start < 1000);
29        /*repeat until enough time has elapsed*/
30
31        duration = ((double)(clock()-start))/CLOCKS_PER_SEC;
32        //여러번 돌렸을 때 걸린 시간
33        duration/=repetitions; //여러번 돌렸을 때 평균 걸린 시간
34        printf("%6d    %9d    %f\n",n, repetitions ,duration);
35        if(n==100)step=100;
36    }
37 }
```

n	repetitions	time
0	6880	0.000000
10	1067	0.000003
20	1590	0.000001
30	517	0.000007
40	597	0.000002
50	372	0.000011
60	254	0.000004
70	189	0.000005
80	138	0.000007
90	105	0.000010
100	85	0.000012
200	25	0.000041
300	8	0.000487
400	7	0.000148
500	5	0.000231
600	4	0.000332
700	3	0.000447
800	2	0.000586
900	1	0.003951
1000	2	0.000911

3.

```

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

int main() {
    clock_t start_time;
    time_t time_start, time_end;
    //sleep(1) 작동 시작 시간
    start_time = clock();
    time_start = time(NULL);

    // 1초 동안 프로세스 일시정지
    sleep(1);

    //sleep(1) 작동 종료 시간
    clock_t end_time = clock();
    time_end = time(NULL);

    double cpu_time_used = ((double) (end_time - start_time)) / CLOCKS_PER_SEC;
    double real_time_used = difftime(time_end, time_start);

    printf("clock(): %f s\n", cpu_time_used);
    printf("time(): %f s\n", real_time_used);
    /*
    clock()은 프로세스가 시작한 시간부터 실행되는 동안의 시간을 세기 때문에
    sleep(1)이 실행되는 동안에는 시간을 세지 않는 반면,
    time()은 시스템 시간이기 때문에 프로세스가 일시정지해도 계속해서 시간을 센다.
    */
    return 0;
}
```

```
clock(): 0.000015 s
time(): 1.000000 s
```