

## 자료구조 실습과제 17

- 솔루션 및 프로젝트 명칭 : Proj\_17\_이름이니셜
- 제출방법 : 아래 문제를 해결하기 위한 프로그램을 구현한 후 컴파일 및 실행한 후, 오류가 없으면 메뉴에서 솔루션 정리를 수행한 후 윈도우 탐색기에서 솔루션 폴더를 찾아 압축하여 E-class에 올림

### 문제 1) 선택정렬, 삽입정렬, 버블정렬, 쉘정렬, 합병정렬, 퀵정렬, 힙정렬, 기수정렬 구현

정수를 대상으로 다양한 정렬방법을 구현하여 데이터 수가 매우 커질 경우 각 정렬방법의 시간복잡도를 계산하시오. 정렬에 사용되는 데이터는 main 함수 내에서 random seed를 설정하는 srand()와 rand()를 이용하여 data 배열에 저장한 후, 동일한 크기의 list에 복사한 후 각각의 알고리즘을 테스트한다. 각 알고리즘을 수행하는 데 걸리는 시간은 각 함수를 호출하기 전과 후에 clock() 함수를 호출하여 두 시간 차이를 출력한다. 먼저 100개의 데이터를 생성하여 각 알고리즘이 정상적으로 오름차순으로 정렬이 되는지 확인한다. 알고리즘 동작이 정상적임을 확인한 경우 데이터 사이즈를 10,000, 50,000, 100,000으로 설정하여 각 알고리즘이 동작하는 데 걸리는 시간을 출력한다. 아래 main() 함수의 코드와 실행 예를 참고하시오.

```
#define SIZE 100 // 데이터크기를 변경하며 테스트
int main(void)
{
    int i, s_time, e_time;
    int* data = (int*)malloc(sizeof(int) * SIZE); // 각 정렬을 동일한 데이터로 테스트
    하기 위한 원데이터
    int* list = (int*)malloc(sizeof(int) * SIZE); // 원데이터 -> 정렬된 데이터
    sorted = (int*)malloc(sizeof(int) * SIZE); //병합정렬을 위한 추가 메모리로 글로벌
    변수로 선언

    if (data == NULL || list == NULL || sorted == NULL) {
        printf("데이터 저장을 위한 메모리 할당 오류\n"); exit(-1);
    }

    printf("정렬 기법 비교 \n");
    printf("Data 크기 : %d \n\n", SIZE);
    srand(100);
    for (i = 0; i < SIZE; i++) // 난수 생성 및 출력
        data[i] = rand(); // 난수 발생

    memcpy(list, data, sizeof(int) * SIZE);
    s_time = clock();
    selection_sort(list, SIZE); // 선택정렬 호출
    e_time = clock();
    printf("선택정렬 시간 : %d\n", e_time - s_time);
```

```

if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
insertion_sort(list, SIZE);
e_time = clock();
printf("삽입정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
bubble_sort(list, SIZE);
e_time = clock();
printf("버블정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
shell_sort(list, SIZE);
e_time = clock();
printf("셸정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
merge_sort(list, 0, SIZE-1);
e_time = clock();
printf("병합정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

```

```

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
quick_sort(list, 0, SIZE - 1);
e_time = clock();
printf("퀵정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
heap_sort(list, SIZE);
e_time = clock();
printf("힙정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

memcpy(list, data, sizeof(int) * SIZE);
s_time = clock();
radix_sort(list, SIZE); // 기수정렬 호출
e_time = clock();
printf("기수정렬 시간 : %d\n", e_time - s_time);
if (SIZE <= 100) {
    for (i = 0; i < SIZE; i++)
        printf("%d ", list[i]);
    printf("\n\n");
}

free(data); free(list); free(sorted);

return 0;
}

```

실행 예 : (data size == 100)

```
Microsoft Visual Studio 디버그 콘솔

정렬 기법 비교
Data 크기 : 100

선택정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

삽입정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

버블정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

셸정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

병합정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

퀵정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

힉정렬 시간 : 0
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760

기수정렬 시간 : 6
310 365 1031 1216 1435 1580 1598 1702 1825 1983 1989 2314 2828 3768 4195 4243 4482 5347 5415 5492 5629 5637 6401 6470 67
25 7072 7121 7628 7851 8834 8973 9540 9960 10254 10524 10633 11254 11271 11409 11529 11605 11867 11875 12136 12245 12442
13018 13026 13079 14203 15177 15274 15489 15646 15810 15921 15961 16241 16513 16704 16867 17112 17251 17364 17391 17635
17688 17995 18121 18834 18993 19207 19574 19867 20443 22030 22229 22275 22286 22442 23209 23471 23796 23830 24504 24696
24698 25262 25857 25907 25981 28727 28924 28980 29816 29982 31753 32505 32578 32760
```

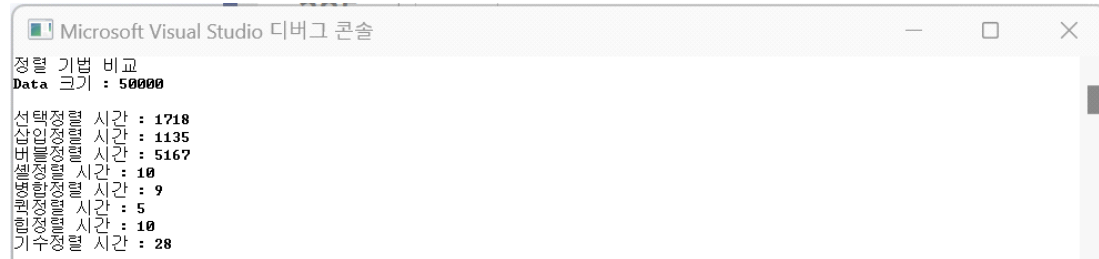
실행 예 : (data size == 10000)

```
Microsoft Visual Studio 디버그 콘솔

정렬 기법 비교
Data 크기 : 10000

선택정렬 시간 : 74
삽입정렬 시간 : 45
버블정렬 시간 : 152
셸정렬 시간 : 2
병합정렬 시간 : 2
퀵정렬 시간 : 1
힉정렬 시간 : 2
기수정렬 시간 : 10
```

실행 예 : (data size == 50000)

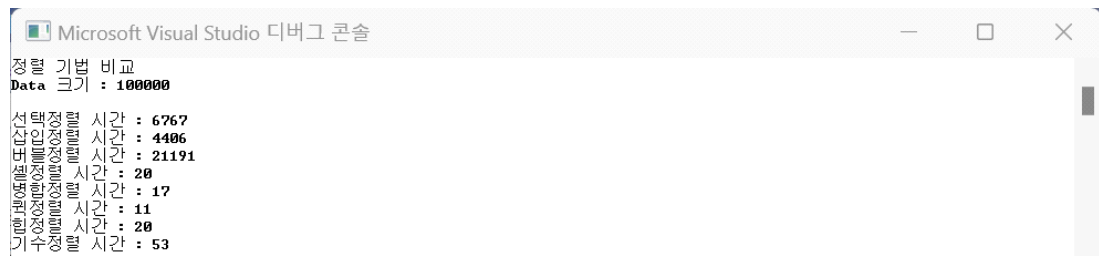


```
Microsoft Visual Studio 디버그 콘솔

정렬 기법 비교
Data 크기 : 50000

선택정렬 시간 : 1718
삽입정렬 시간 : 1135
버블정렬 시간 : 5167
셸정렬 시간 : 10
병합정렬 시간 : 9
퀵정렬 시간 : 5
힙정렬 시간 : 10
기수정렬 시간 : 28
```

실행 예 : (data size == 100000)



```
Microsoft Visual Studio 디버그 콘솔

정렬 기법 비교
Data 크기 : 100000

선택정렬 시간 : 6767
삽입정렬 시간 : 4406
버블정렬 시간 : 21191
셸정렬 시간 : 20
병합정렬 시간 : 17
퀵정렬 시간 : 11
힙정렬 시간 : 20
기수정렬 시간 : 53
```