REPORT

과 목 명 : 자료구조

담담교수 : 황두성 교수님

소　　속 : 소프트웨어학과

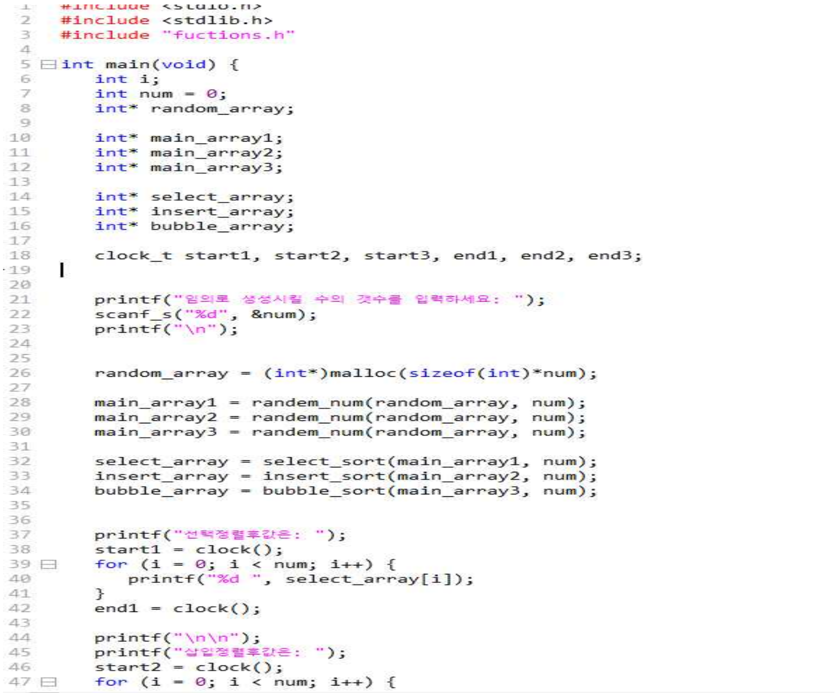
학　　번 : 32151671

이　　름 : 박민혁

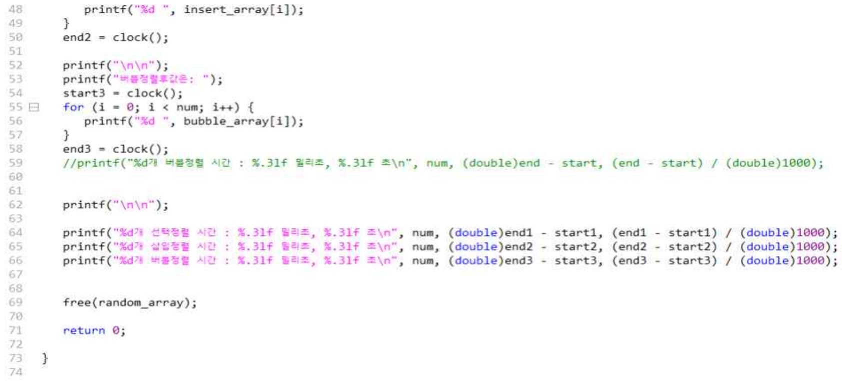
단국대학교
Dankook University

정렬

main.c

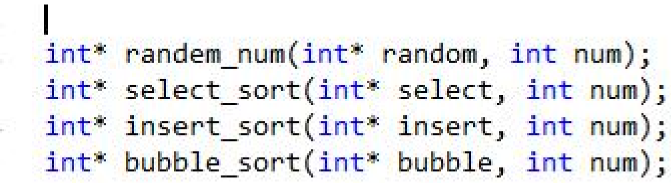


```c
#include <stdio.h>
#include <stdlib.h>
#include "fuctions.h"

int main(void) {
    int i;
    int num = 0;
    int* random_array;

    int* main_array1;
    int* main_array2;
    int* main_array3;

    int* select_array;
    int* insert_array;
    int* bubble_array;

    clock_t start1, start2, start3, end1, end2, end3;


    printf("임의로 생성시킬 수의 갯수를 입력하세요: ");
    scanf_s("%d", &num);
    printf("\n");


    random_array = (int*)malloc(sizeof(int)*num);

    main_array1 = randem_num(random_array, num);
    main_array2 = randem_num(random_array, num);
    main_array3 = randem_num(random_array, num);

    select_array = select_sort(main_array1, num);
    insert_array = insert_sort(main_array2, num);
    bubble_array = bubble_sort(main_array3, num);


    printf("선택정렬후값은: ");
    start1 = clock();
    for (i = 0; i < num; i++) {
        printf("%d ", select_array[i]);
    }
    end1 = clock();

    printf("\n\n");
    printf("삽입정렬후값은: ");
    start2 = clock();
    for (i = 0; i < num; i++) {
```



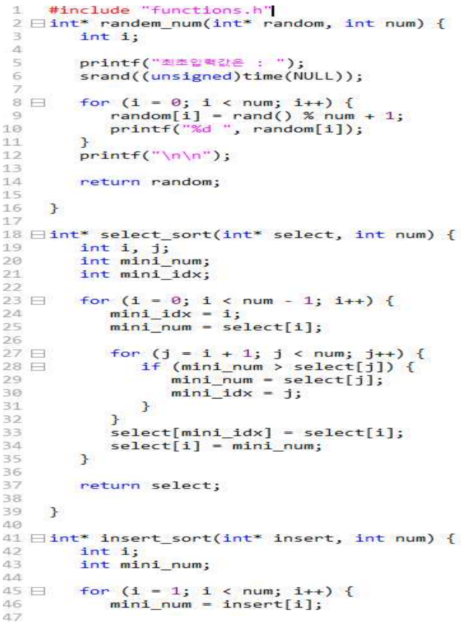
```c
        printf("%d ", insert_array[i]);
    }
    end2 = clock();

    printf("\n\n");
    printf("버블정렬후값은: ");
    start3 = clock();
    for (i = 0; i < num; i++) {
        printf("%d ", bubble_array[i]);
    }
    end3 = clock();
    //printf("%d개 버블정렬 시간 : %.31f 밀리초, %.31f 초\n", num, (double)end - start, (end - start) / (double)1000);


    printf("\n\n");

    printf("%d개 선택정렬 시간 : %.31f 밀리초, %.31f 초\n", num, (double)end1 - start1, (end1 - start1) / (double)1000);
    printf("%d개 삽입정렬 시간 : %.31f 밀리초, %.31f 초\n", num, (double)end2 - start2, (end2 - start2) / (double)1000);
    printf("%d개 버블정렬 시간 : %.31f 밀리초, %.31f 초\n", num, (double)end3 - start3, (end3 - start3) / (double)1000);


    free(random_array);

    return 0;

}
```

functions.h

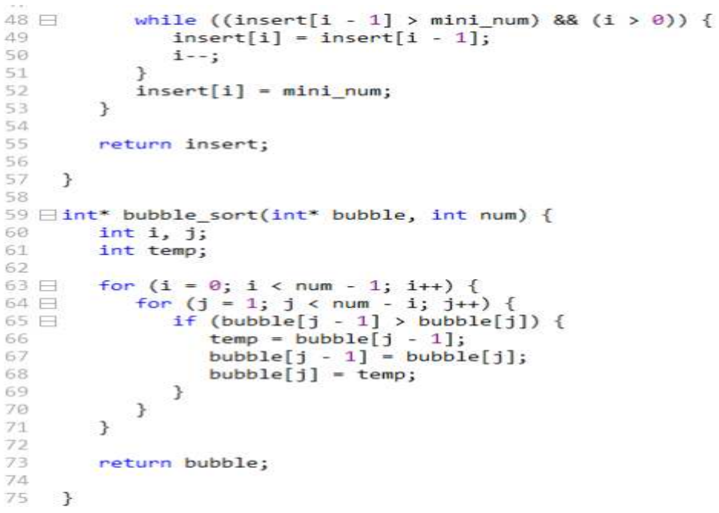


```c
int* randem_num(int* random, int num);
int* select_sort(int* select, int num);
int* insert_sort(int* insert, int num);
int* bubble_sort(int* bubble, int num);
```

functions.c

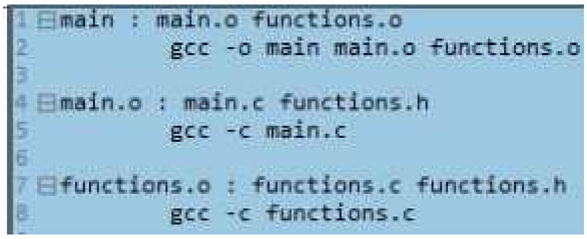


```c
#include "functions.h"
int* randem_num(int* random, int num) {
    int i;

    printf("최초입력값은 : ");
    srand((unsigned)time(NULL));

    for (i = 0; i < num; i++) {
        random[i] = rand() % num + 1;
        printf("%d ", random[i]);
    }
    printf("\n\n");

    return random;

}

int* select_sort(int* select, int num) {
    int i, j;
    int mini_num;
    int mini_idx;

    for (i = 0; i < num - 1; i++) {
        mini_idx = i;
        mini_num = select[i];

        for (j = i + 1; j < num; j++) {
            if (mini_num > select[j]) {
                mini_num = select[j];
                mini_idx = j;
            }
        }
        select[mini_idx] = select[i];
        select[i] = mini_num;
    }

    return select;

}

int* insert_sort(int* insert, int num) {
    int i;
    int mini_num;

    for (i = 1; i < num; i++) {
        mini_num = insert[i];

```



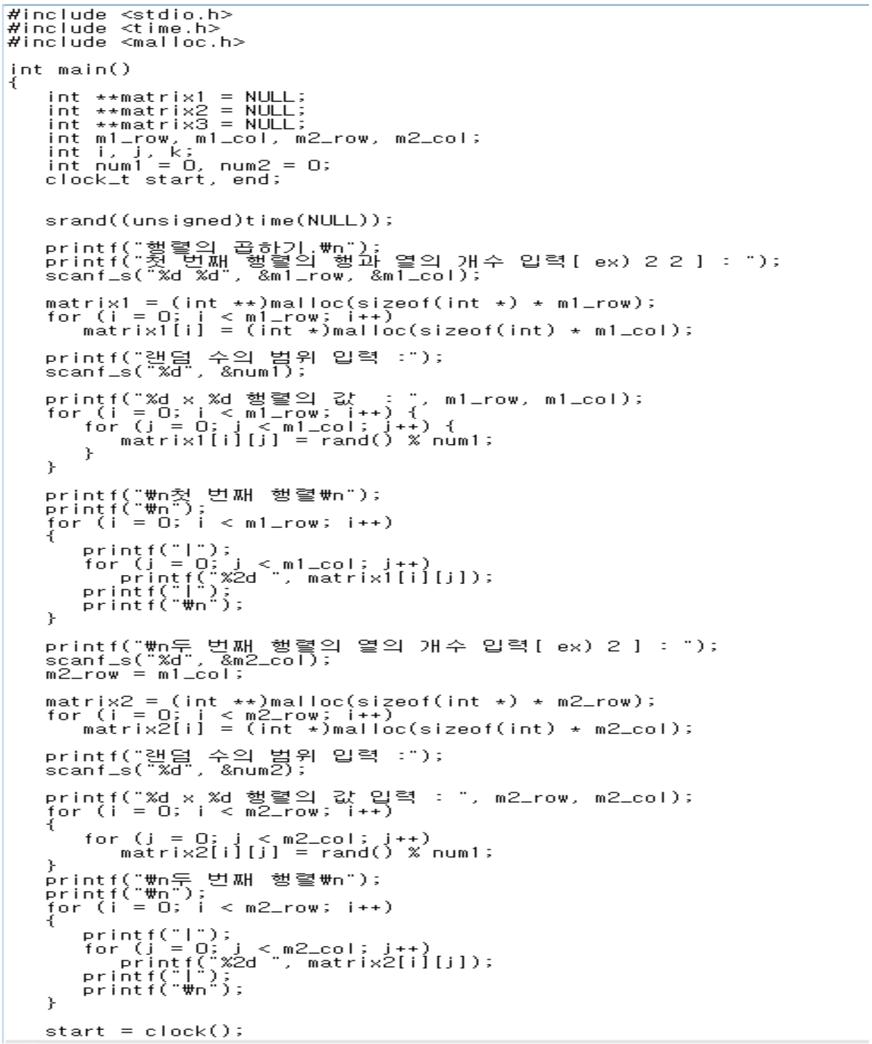
```c
            while ((insert[i - 1] > mini_num) && (i > 0)) {
                insert[i] = insert[i - 1];
                i--;
            }
            insert[i] = mini_num;
        }

        return insert;

    }

    int* bubble_sort(int* bubble, int num) {
        int i, j;
        int temp;

        for (i = 0; i < num - 1; i++) {
            for (j = 1; j < num - i; j++) {
                if (bubble[j - 1] > bubble[j]) {
                    temp = bubble[j - 1];
                    bubble[j - 1] = bubble[j];
                    bubble[j] = temp;
                }
            }
        }

        return bubble;

    }
```

makefile

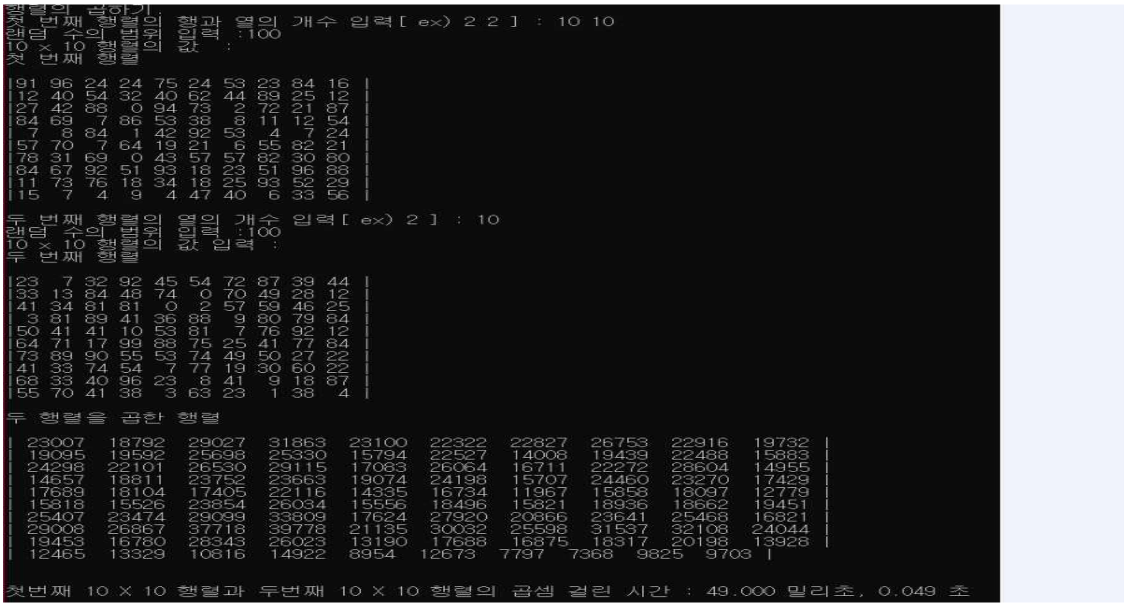


```makefile
main : main.o functions.o
	gcc -o main main.o functions.o

main.o : main.c functions.h
	gcc -c main.c

functions.o : functions.c functions.h
	gcc -c functions.c
```

행렬



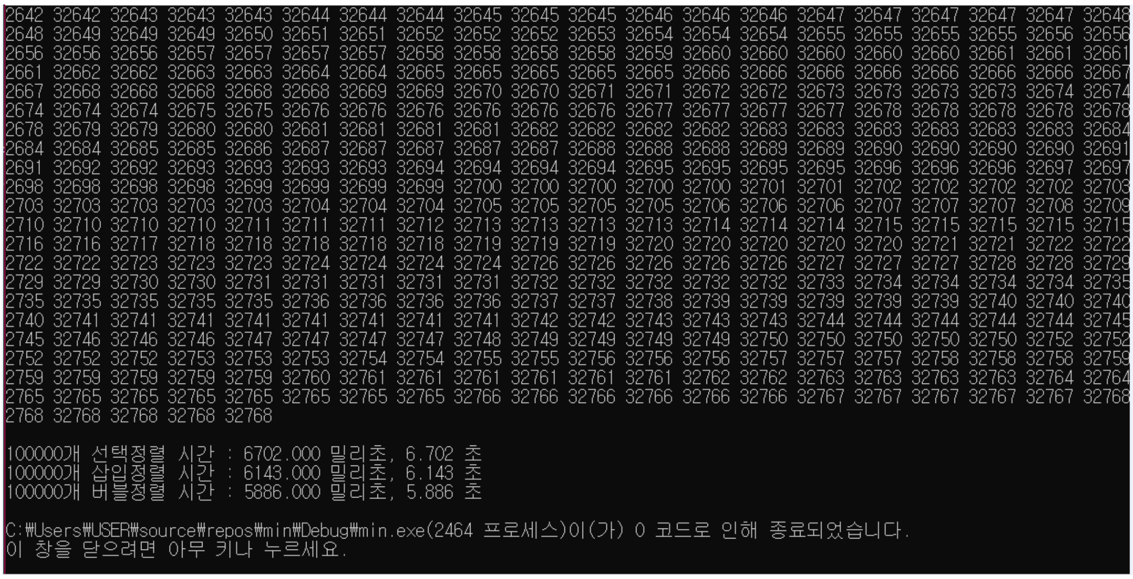
```
#include <stdio.h>
#include <time.h>
#include <malloc.h>

int main()
{
    int **matrix1 = NULL;
    int **matrix2 = NULL;
    int **matrix3 = NULL;
    int m1_row, m1_col, m2_row, m2_col;
    int i, j, k;
    int num1 = 0, num2 = 0;
    clock_t start, end;


    srand((unsigned)time(NULL));

    printf("행렬의 곱하기.₩n");
    printf("첫 번째 행렬의 행과 열의 개수 입력[ ex) 2 2 ] : ");
    scanf_s("%d %d", &m1_row, &m1_col);

    matrix1 = (int **)malloc(sizeof(int *) * m1_row);
    for (i = 0; i < m1_row; i++)
        matrix1[i] = (int *)malloc(sizeof(int) * m1_col);

    printf("랜덤 수의 범위 입력 :");
    scanf_s("%d", &num1);

    printf("%d x %d 행렬의 값  : ", m1_row, m1_col);
    for (i = 0; i < m1_row; i++) {
        for (j = 0; j < m1_col; j++) {
            matrix1[i][j] = rand() % num1;
        }
    }

    printf("₩n첫 번째 행렬₩n");
    printf("₩n");
    for (i = 0; i < m1_row; i++)
    {
        printf("|");
        for (j = 0; j < m1_col; j++)
            printf("%2d ", matrix1[i][j]);
        printf("|");
        printf("₩n");
    }

    printf("₩n두 번째 행렬의 열의 개수 입력[ ex) 2 ] : ");
    scanf_s("%d", &m2_col);
    m2_row = m1_col;

    matrix2 = (int **)malloc(sizeof(int *) * m2_row);
    for (i = 0; i < m2_row; i++)
        matrix2[i] = (int *)malloc(sizeof(int) * m2_col);

    printf("랜덤 수의 범위 입력 :");
    scanf_s("%d", &num2);

    printf("%d x %d 행렬의 값 입력 : ", m2_row, m2_col);
    for (i = 0; i < m2_row; i++)
    {
        for (j = 0; j < m2_col; j++)
            matrix2[i][j] = rand() % num1;
    }
    printf("₩n두 번째 행렬₩n");
    printf("₩n");
    for (i = 0; i < m2_row; i++)
    {
        printf("|");
        for (j = 0; j < m2_col; j++)
            printf("%2d ", matrix2[i][j]);
        printf("|");
        printf("₩n");
    }

    start = clock();
    matrix3 = (int **)malloc(sizeof(int *) * m1_row);
    for (i = 0; i < m1_row; i++)
        matrix3[i] = (int *)calloc(m2_col, sizeof(int));

    for (i = 0; i < m1_row; i++)
    {
        for (j = 0; j < m2_col; j++)
        {
            for (k = 0; k < m1_col; k++)
                matrix3[i][j] += matrix1[i][k] * matrix2[k][j];
        }
    }


    printf("₩n두 행렬을 곱한 행렬₩n");
    printf("₩n");

    for (i = 0; i < m1_row; i++) {
        printf("|");
        for (j = 0; j < m2_col; j++) {
            printf(" %2d ", matrix3[i][j]);
        }
        printf("|");
        printf("₩n");
    }
    end = clock();

    printf("₩n₩n");
    printf("첫번째 %d X %d 행렬과 두번째 %d X %d 행렬의 곱셈 걸린 시간 : %.3lf 밀리초, %.3lf 초₩n", m1_row, m1_col, m2_row, m2_col, (double)end - start, (end - start) / (double)1000);

    for (i = 0; i < m1_row; i++) {
        free(matrix1[i]);
        free(matrix3[i]);
    }
    for (i = 0; i < m2_row; i++)
        free(matrix2[i]);



    free(matrix1);
    free(matrix2);
    free(matrix3);

    return 0;
}
```



```
행렬의 곱하기
첫 번째 행렬의 행과 열의 개수 입력[ ex) 2 2 ] : 10 10
랜덤 수의 범위 입력 :100
10 x 10 행렬의 값  :
첫 번째 행렬

|91 96 24 24 75 24 53 23 84 16 |
|12 40 54 32 40 62 44 89 25 12 |
|27 42 88  0 94 73  2 72 21 87 |
|84 69  7 86 53 38  8 11 12 54 |
| 7  8 84  1 42 92 53  4  7 24 |
|57 70  7 64 19 21  6 55 82 21 |
|78 31 69  0 43 57 57 82 30 80 |
|84 67 92 51 93 18 23 51 96 88 |
|11 73 76 18 34 18 25 93 52 29 |
|15  7  4  9  4 47 40  6 33 56 |

두 번째 행렬의 열의 개수 입력[ ex) 2 ] : 10
랜덤 수의 범위 입력 :100
10 x 10 행렬의 값 입력 :
두 번째 행렬

|23  7 32 92 45 54 72 87 39 44 |
|33 13 84 48 74  0 70 49 28 12 |
|41 34 81 81  0  2 57 59 46 25 |
| 3 81 89 41 36 88  9 80 79 84 |
|50 41 41 10 53 81  7 76 92 12 |
|64 71 17 99 88 75 25 41 77 84 |
|73 89 90 55 53 74 49 50 27 22 |
|41 33 74 54  7 77 19 30 60 22 |
|68 33 40 96 23  8 41  9 18 87 |
|55 70 41 38  3 63 23  1 38  4 |

두 행렬을 곱한 행렬

|  23007   18792   29027   31863   23100   22322   22827   26753   22916   19732  |
|  19095   19592   25698   25330   15794   22527   14008   19439   22488   15883  |
|  24298   22101   26530   29115   17083   26064   16711   22272   28604   14955  |
|  14657   18811   23752   23663   19074   24198   15707   24460   23270   17429  |
|  17689   18104   17405   22116   14335   16734   11967   15858   18097   12779  |
|  15818   15526   23854   26034   15556   18496   15821   18936   18662   19451  |
|  25407   23474   29099   33809   17624   27920   20866   23641   25468   16821  |
|  29008   26867   37718   39778   21135   30032   25598   31537   32108   24044  |
|  19453   16780   28343   26023   13190   17688   16875   18317   20198   13928  |
|  12465   13329   10816   14922   8954   12673   7797   7368   9825   9703 |


첫번째 10 X 10 행렬과 두번째 10 X 10 행렬의 곱셈 걸린 시간 : 49.000 밀리초, 0.049 초
```

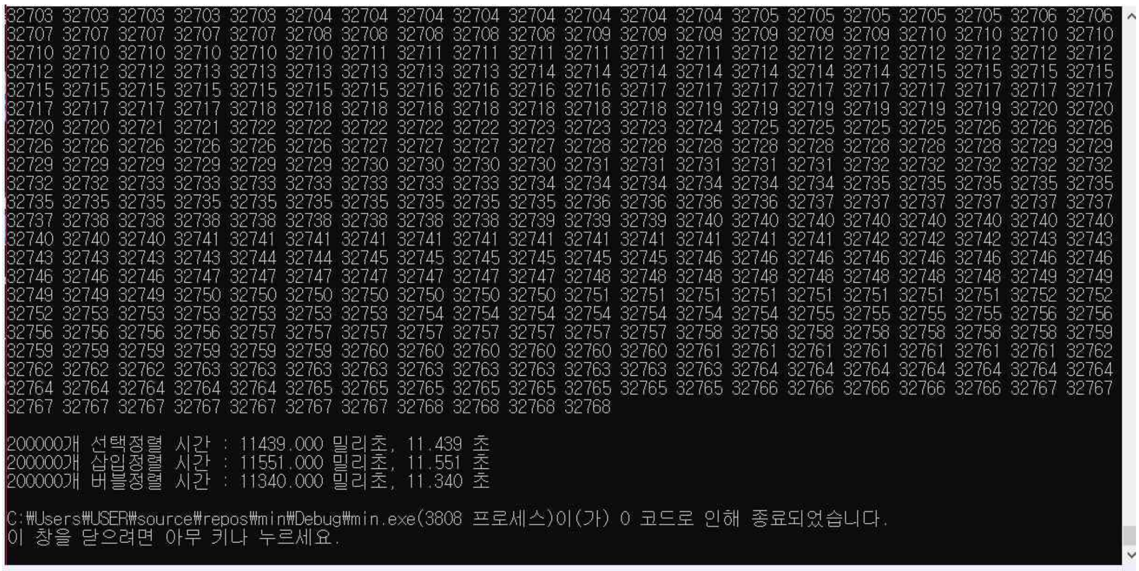
정렬결과

10만



```
2642 32642 32643 32643 32643 32644 32644 32644 32645 32645 32645 32646 32646 32646 32647 32647 32647 32647 32647 32648
2648 32649 32649 32649 32650 32651 32651 32652 32652 32652 32653 32654 32654 32654 32655 32655 32655 32655 32656 32656
2656 32656 32656 32657 32657 32657 32657 32658 32658 32658 32658 32659 32660 32660 32660 32660 32660 32661 32661 32661
2661 32662 32662 32663 32663 32664 32664 32665 32665 32665 32665 32665 32666 32666 32666 32666 32666 32666 32666 32667
2667 32668 32668 32668 32668 32668 32669 32669 32670 32670 32671 32671 32672 32672 32673 32673 32673 32673 32674 32674
2674 32674 32674 32675 32675 32676 32676 32676 32676 32676 32676 32677 32677 32677 32677 32678 32678 32678 32678 32678
2678 32679 32679 32680 32680 32681 32681 32681 32681 32682 32682 32682 32682 32683 32683 32683 32683 32683 32683 32684
2684 32684 32685 32685 32686 32687 32687 32687 32687 32687 32688 32688 32688 32689 32689 32690 32690 32690 32690 32691
2691 32692 32692 32693 32693 32693 32693 32694 32694 32694 32694 32695 32695 32695 32695 32696 32696 32696 32697 32697
2698 32698 32698 32698 32699 32699 32699 32699 32700 32700 32700 32700 32700 32701 32701 32702 32702 32702 32702 32703
2703 32703 32703 32703 32703 32704 32704 32704 32705 32705 32705 32705 32706 32706 32706 32707 32707 32707 32708 32709
2710 32710 32710 32710 32711 32711 32711 32712 32713 32713 32713 32713 32714 32714 32714 32715 32715 32715 32715 32715
2716 32716 32717 32718 32718 32718 32718 32718 32719 32719 32719 32720 32720 32720 32720 32720 32721 32721 32722 32722
2722 32722 32723 32723 32723 32724 32724 32724 32724 32726 32726 32726 32726 32726 32727 32727 32727 32728 32728 32729
2729 32729 32730 32730 32731 32731 32731 32731 32731 32732 32732 32732 32732 32732 32733 32734 32734 32734 32734 32735
2735 32735 32735 32735 32735 32736 32736 32736 32736 32737 32737 32738 32739 32739 32739 32739 32739 32740 32740 32740
2740 32741 32741 32741 32741 32741 32741 32741 32741 32742 32742 32743 32743 32743 32744 32744 32744 32744 32744 32745
2745 32746 32746 32746 32747 32747 32747 32747 32748 32749 32749 32749 32749 32750 32750 32750 32750 32750 32752 32752
2752 32752 32752 32753 32753 32753 32754 32754 32755 32755 32756 32756 32756 32757 32757 32757 32758 32758 32758 32759
2759 32759 32759 32759 32759 32760 32761 32761 32761 32761 32761 32761 32762 32762 32763 32763 32763 32763 32764 32764
2765 32765 32765 32765 32765 32765 32765 32765 32766 32766 32766 32766 32766 32766 32767 32767 32767 32767 32767 32768
2768 32768 32768 32768 32768

100000개 선택정렬 시간 : 6702.000 밀리초, 6.702 초
100000개 삽입정렬 시간 : 6143.000 밀리초, 6.143 초
100000개 버블정렬 시간 : 5886.000 밀리초, 5.886 초

C:₩Users₩USER₩source₩repos₩min₩Debug₩min.exe(2464 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

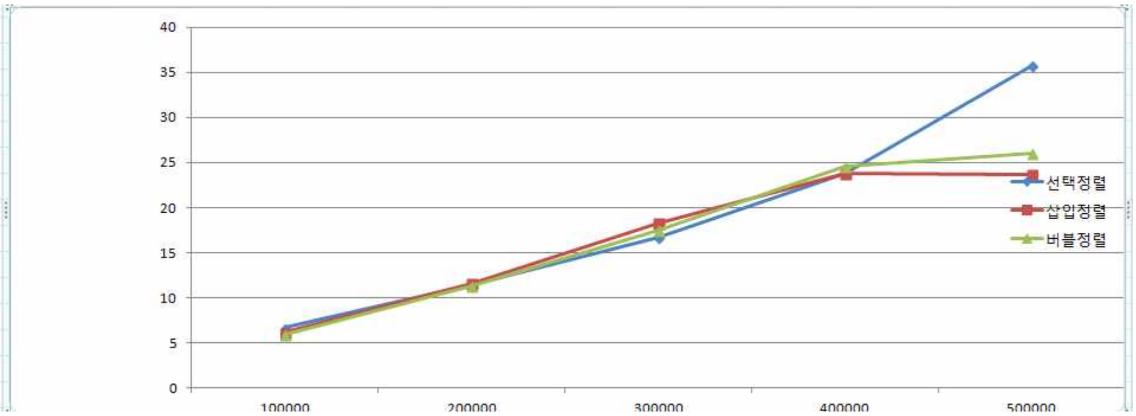
20만



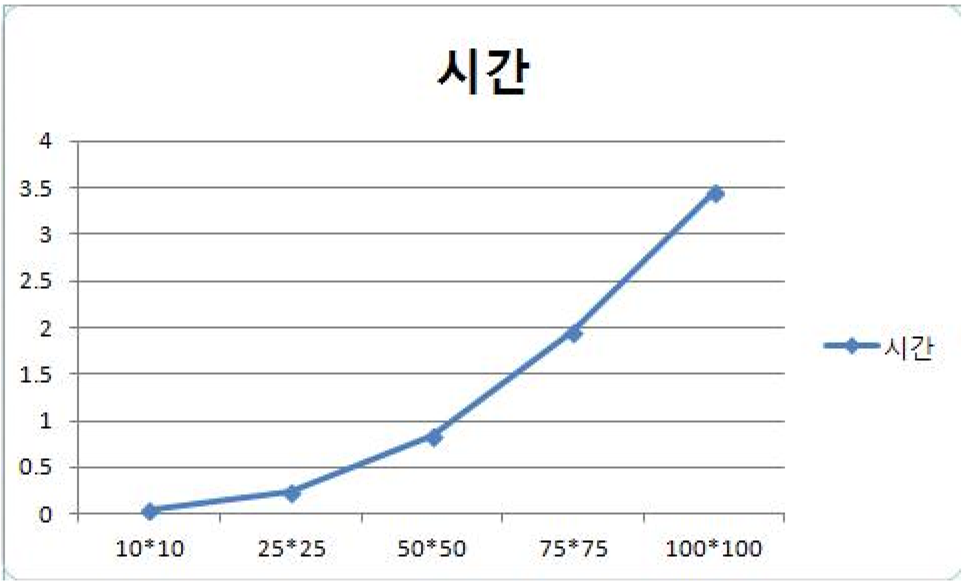
```
32703 32703 32703 32703 32703 32704 32704 32704 32704 32704 32704 32704 32704 32705 32705 32705 32705 32705 32706 32706
32707 32707 32707 32707 32707 32708 32708 32708 32708 32708 32709 32709 32709 32709 32709 32709 32710 32710 32710 32710
32710 32710 32710 32710 32710 32710 32711 32711 32711 32711 32711 32711 32711 32712 32712 32712 32712 32712 32712 32712
32712 32712 32712 32713 32713 32713 32713 32713 32713 32714 32714 32714 32714 32714 32714 32714 32715 32715 32715 32715
32715 32715 32715 32715 32715 32715 32715 32716 32716 32716 32716 32717 32717 32717 32717 32717 32717 32717 32717 32717
32717 32717 32717 32717 32718 32718 32718 32718 32718 32718 32718 32718 32719 32719 32719 32719 32719 32719 32720 32720
32720 32720 32721 32721 32722 32722 32722 32722 32722 32723 32723 32723 32724 32725 32725 32725 32725 32726 32726 32726
32726 32726 32726 32726 32726 32726 32727 32727 32727 32727 32728 32728 32728 32728 32728 32728 32728 32728 32729 32729
32729 32729 32729 32729 32729 32729 32730 32730 32730 32730 32731 32731 32731 32731 32731 32732 32732 32732 32732 32732
32732 32732 32733 32733 32733 32733 32733 32733 32733 32734 32734 32734 32734 32734 32734 32735 32735 32735 32735 32735
32735 32735 32735 32735 32735 32735 32735 32735 32735 32735 32736 32736 32736 32736 32737 32737 32737 32737 32737 32737
32737 32738 32738 32738 32738 32738 32738 32738 32738 32739 32739 32739 32740 32740 32740 32740 32740 32740 32740 32740
32740 32740 32740 32741 32741 32741 32741 32741 32741 32741 32741 32741 32741 32741 32741 32742 32742 32742 32743 32743
32743 32743 32743 32743 32744 32744 32745 32745 32745 32745 32745 32745 32746 32746 32746 32746 32746 32746 32746 32746
32746 32746 32746 32747 32747 32747 32747 32747 32747 32747 32748 32748 32748 32748 32748 32748 32748 32748 32749 32749
32749 32749 32749 32750 32750 32750 32750 32750 32750 32750 32751 32751 32751 32751 32751 32751 32751 32751 32752 32752
32752 32753 32753 32753 32753 32753 32753 32754 32754 32754 32754 32754 32754 32754 32755 32755 32755 32755 32756 32756
32756 32756 32756 32756 32757 32757 32757 32757 32757 32757 32757 32757 32758 32758 32758 32758 32758 32758 32758 32759
32759 32759 32759 32759 32759 32759 32760 32760 32760 32760 32760 32760 32761 32761 32761 32761 32761 32761 32761 32762
32762 32762 32762 32763 32763 32763 32763 32763 32763 32763 32763 32763 32763 32764 32764 32764 32764 32764 32764 32764
32764 32764 32764 32764 32764 32765 32765 32765 32765 32765 32765 32765 32765 32766 32766 32766 32766 32766 32767 32767
32767 32767 32767 32767 32767 32767 32767 32768 32768 32768 32768

200000개 선택정렬 시간 : 11439.000 밀리초, 11.439 초
200000개 삽입정렬 시간 : 11551.000 밀리초, 11.551 초
200000개 버블정렬 시간 : 11340.000 밀리초, 11.340 초

C:₩Users₩USER₩source₩repos₩min₩Debug₩min.exe(3808 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

시간그래프

정렬



행렬



<조건 : random 데이터의 범위 10 이하>

첫번째 10 X 10 행렬과 두번째 10 X 10 행렬의 곱셈 걸린 시간 : 39.000 밀리초, 0.039 초
첫번째 25 X 25 행렬과 두번째 25 X 25 행렬의 곱셈 걸린 시간 : 223.000 밀리초, 0.223 초
첫번째 50 X 50 행렬과 두번째 50 X 50 행렬의 곱셈 걸린 시간 : 720.000 밀리초, 0.720 초
첫번째 75 X 75 행렬과 두번째 75 X 75 행렬의 곱셈 걸린 시간 : 1354.000 밀리초, 1.354 초
첫번째 100 X 100 행렬과 두번째 100 X 100 행렬의 곱셈 걸린 시간 : 2631.000 밀리초, 2.631 초