Code Optimization

sort version 1 (sort1.c)

```
void sort(int *a, int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = n - 1; j > i; j--) {
            if (a[i] > a[j]) {
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
}
```

main program (main.c)

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

main() {
    int i, startTime, N = 100000;
    int *a = (int *) malloc(N * sizeof(int));
    srand(7913);
    for (i = 0; i < N; i++)
        a[i] = rand();
    startTime = clock();
    sort(a, N);
    printf("Time used = %0.1f sec.\n",
           (float) (clock() - startTime) /
                   CLOCKS_PER_SEC);
}
```

Compilation

```
gcc –o main main.c sort1.c
```

Run

```
./main
```

Output

Time used = 30.9 sec.

MacBook Pro, 2.53 GHz, Intel Core 2 Duo

sort version 2

(faster test in inner loop)

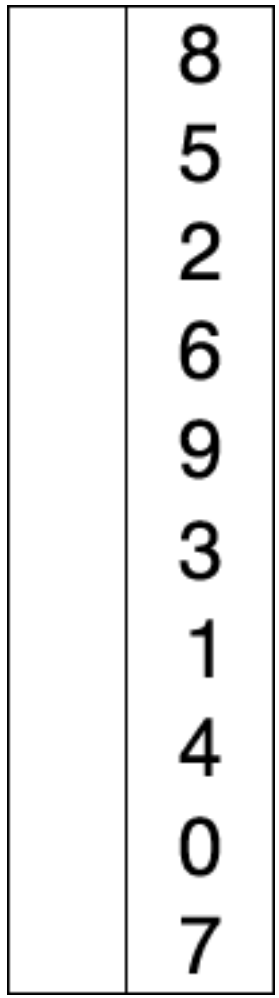
```
void sort(int *a, int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        int temp = a[i];
        for (j = n - 1; j > i; j--) {
            if (temp > a[j]) {
                a[i] = a[j];
                a[j] = temp;
                temp = a[i];
            }
        }
    }
}
```

Time used = 16.2 sec.

sort version 3

(avoiding unnecessary swaps in inner loop)

```
void sort(int *a, int n) {
    int i, j, ajmin, jmin;
    for (i = 0; i < n; i++) {
        ajmin = a[jmin = i];
        for (j = n - 1; j > i; j--)
            if (a[j] < ajmin)
                ajmin = a[jmin = j];
        a[jmin] = a[i];
        a[i] = ajmin;
    }
}
```



Time used = 16.1 sec.

sort version 4

(unwinding the inner loop)

```
for (j = n - 1 ; j >= i + 10; j -= 10) {
    if (a[j] < ajmin) ajmin = a[jmin = j];
    if (a[j - 1] < ajmin) ajmin = a[jmin = j - 1];
    if (a[j - 2] < ajmin) ajmin = a[jmin = j - 2];
    if (a[j - 3] < ajmin) ajmin = a[jmin = j - 3];
    if (a[j - 4] < ajmin) ajmin = a[jmin = j - 4];
    if (a[j - 5] < ajmin) ajmin = a[jmin = j - 5];
    if (a[j - 6] < ajmin) ajmin = a[jmin = j - 6];
    if (a[j - 7] < ajmin) ajmin = a[jmin = j - 7];
    if (a[j - 8] < ajmin) ajmin = a[jmin = j - 8];
    if (a[j - 9] < ajmin) ajmin = a[jmin = j - 9];
}
for ( ; j > i; j--)
    if (a[j] < ajmin)
        ajmin = a[jmin = j];
```

Time used = 9.0 sec.

Compiler optimization

```
gcc –o main –O3 main.c sort.c
```

	No optimization	Optimization	Java
Version 1	30.9 sec.	10.8 sec.	21.0 sec.
Version 2	16.2 sec.	11.7 sec.	15.6 sec.
Version 3	16.1 sec.	4.8 sec.	12.2 sec.
Version 4	9.0 sec.	8.0 sec.	4.6 sec.