

# Project 1

## Section 1.3 – Sorting Algorithms

### Rainbow Martin

#### Hardware Specs:

- CPU: 11<sup>th</sup> Gen Intel(R) Core(TM) i7-1195G7 @2.90GHz, 4 cores
- 8 GB of RAM
- 475GB of Storage
- GPU Intel(R) Iris(R) Xe Graphics, 128 MB
- OS: Windows 11

$n$	Time taken by Selection sort (in seconds)	Time taken by Counting sort (in seconds)
$10^2$	0.000000	0.000000
$10^3$	0.002000	0.000000
$10^4$	0.280000	0.001000
$10^5$	29.365000	0.002000
$10^6$	2293.487000	0.016000
$10^7$	*	0.145000
$10^8$	*	2.004000
$10^9$	*	73.729000

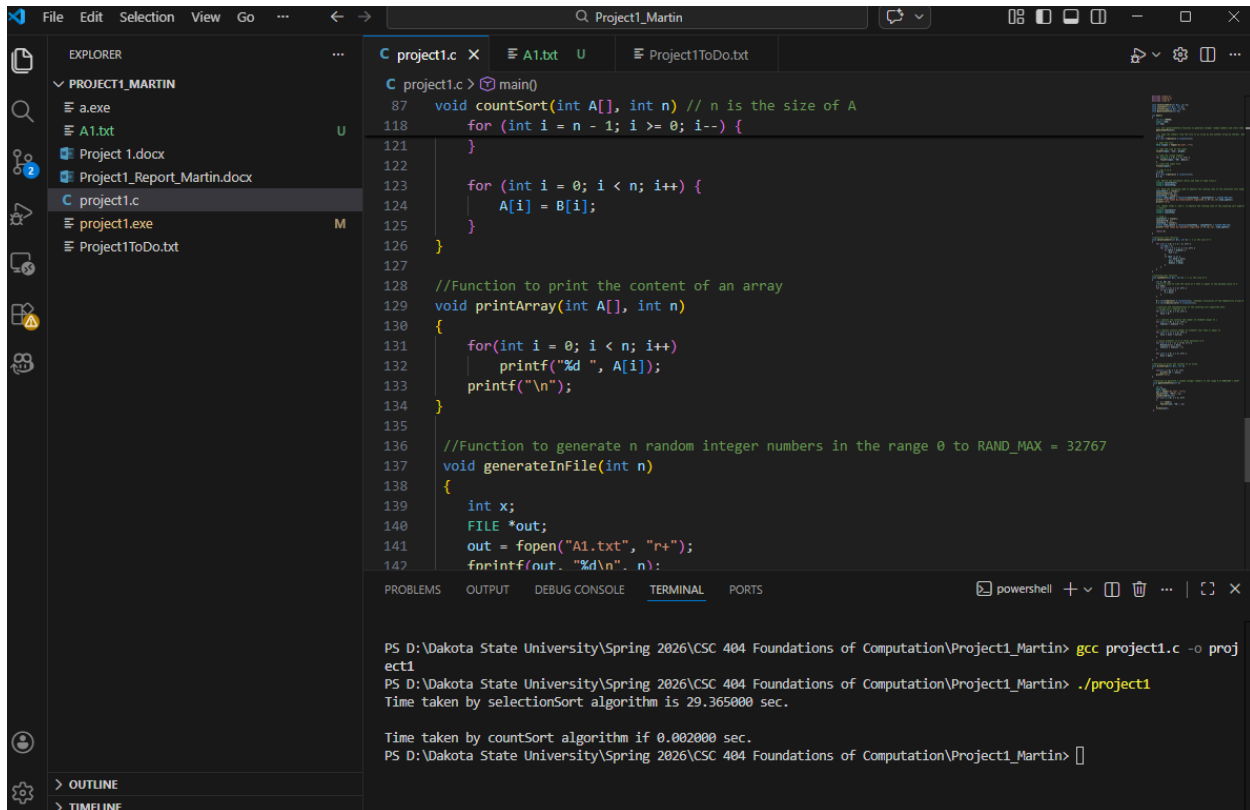
\* Algorithm time growth is exponential – will take days to run

Looking at my output, it is obvious that the Selection Sort algorithm's time complexity is  $n^2$  because as the input grows by a factor of 10, the runtime increases rapidly. I originally suspected that an input of size  $10^7$  would take Selection Sort at least 6 hours to complete. However, that was based on multiplying the runtime of  $10^6$  by 10. That is incorrect because the time complexity of the algorithm would not produce linear growth. In addition, I researched the limits of my machine (see specs) and found that running  $10^7$  and above would take longer than a couple days to complete. As you can see from my table, I elected to stop running the Selection Sort algorithm above  $10^6$  due to the lengthy runtime. These results tell me that Selection Sort is not a good candidate for real-world applications, unless said applications are intended for small datasets (i.e. below 10,000). 29 seconds to sort 100,000 isn't terrible, but I can't imagine that many users would be happy with the pause in application response.

The Counting Sort algorithm is much faster and "better" than the Selection Sort algorithm as the input grows because its time complexity is linear based on  $(n + k)$ . The runtime increases less dramatically. I suspect that the higher increases in the  $10^8$  and  $10^9$  input size datasets could be due to possibly higher max values in these arrays compared to the arrays

of lesser input sizes and/or memory issues since I have only 8 GB of RAM. In my research, I found that 16-32 GB of RAM is more adequate for evaluating algorithms with large data sets.

## Selection Sort and Count Sort running on $10^5$ size input



The screenshot displays the Visual Studio Code interface with a C project named 'Project1\_Martin'. The Explorer sidebar on the left shows the project structure, including files like 'a.exe', 'A1.txt', 'Project1.docx', 'Project1\_Report\_Martin.docx', 'project1.c', 'project1.exe', and 'Project1ToDo.txt'. The main editor window shows the source code for 'project1.c', which includes functions for counting sort, printing an array, and generating random integers. The terminal window at the bottom shows the execution of the program, including the compilation command 'gcc project1.c -o project1' and the output of the program, which reports the time taken by the selection sort algorithm as 29.365000 seconds.

```
File Edit Selection View Go ... Project1_Martin
EXPLORER
PROJECT1_MARTIN
  a.exe
  A1.txt
  Project1.docx
  Project1_Report_Martin.docx
  project1.c
  project1.exe
  Project1ToDo.txt

C project1.c X A1.txt U Project1ToDo.txt
C project1.c main()
87 void countSort(int A[], int n) // n is the size of A
118 for (int i = n - 1; i >= 0; i--) {
121 }
122
123 for (int i = 0; i < n; i++) {
124     A[i] = B[i];
125 }
126
127
128 //Function to print the content of an array
129 void printArray(int A[], int n)
130 {
131     for(int i = 0; i < n; i++)
132         printf("%d ", A[i]);
133     printf("\n");
134 }
135
136 //Function to generate n random integer numbers in the range 0 to RAND_MAX = 32767
137 void generateInFile(int n)
138 {
139     int x;
140     FILE *out;
141     out = fopen("A1.txt", "w+");
142     fprintf(out, "%d\n", n);
143
144 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Dakota State University\Spring 2026\CSC 404 Foundations of Computation\Project1_Martin> gcc project1.c -o project1
PS D:\Dakota State University\Spring 2026\CSC 404 Foundations of Computation\Project1_Martin> ./project1
Time taken by selectionSort algorithm is 29.365000 sec.
Time taken by countSort algorithm is 0.002000 sec.
PS D:\Dakota State University\Spring 2026\CSC 404 Foundations of Computation\Project1_Martin>
```

## Selection Sort and Count Sort running on $10^6$ size input

The screenshot shows the Visual Studio Code interface with a C project named 'Project1\_Martin'. The Explorer pane on the left shows the project files: 'a.exe', 'A1.txt', 'Project1.docx', 'project1.c', 'project1.exe', and 'Project1ToDo.txt'. The main editor displays the 'project1.c' file, which contains the following code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void selectionSort(int A[], int n);
6 void countSort(int A[], int n);
7 void printArray(int A[], int n);
8 void generateInFile(int n);
9
10 int main()
11 {
12     int n = 1000000;
13     size_t read;
14     int num;
15
16     //1. Call generateInFile function to generate integer random numbers and store them in a text file
17     generateInFile(n);
18
19     //2. Copy the numbers from the file to an array A1 and another array A2 (A1=A2). Use dynamic allocation to declare A1 and A2
20     int* A;
21     A = (int *)malloc(n * sizeof(int));
22
23     // open the file
24     FILE *inout = fopen("A1.txt", "r");
```

The TERMINAL pane at the bottom shows the execution of the program:

```
PS D:\Dakota State University\Spring 2026\CSC 484 Foundations of Computation\Project1_Martin> gcc project1.c -o project1
PS D:\Dakota State University\Spring 2026\CSC 484 Foundations of Computation\Project1_Martin> ./project1
Time taken by selectionSort algorithm is 2293.487000 sec.
Time taken by countSort algorithm is 0.016000 sec.
PS D:\Dakota State University\Spring 2026\CSC 484 Foundations of Computation\Project1_Martin>
```

## Count Sort running on $10^9$ size input

The screenshot shows the Visual Studio Code interface with the same C project. The main editor displays the 'project1.c' file, which is identical to the previous one, but with the input size changed to  $10^9$ :

```
12 int n = 1000000000;
```

The TERMINAL pane at the bottom shows the execution of the program:

```
PS D:\Dakota State University\Spring 2026\CSC 484 Foundations of Computation\Project1_Martin> gcc project1.c -o project1
PS D:\Dakota State University\Spring 2026\CSC 484 Foundations of Computation\Project1_Martin> ./project1
Time taken by countSort algorithm is 73.729000 sec.
PS D:\Dakota State University\Spring 2026\CSC 484 Foundations of Computation\Project1_Martin>
```