

BLG335E

ANALYSIS OF ALGORITHMS I

CRN:10824

PROJECT 01

INSTRUCTOR:

ZEHRA ÇATALTEPE

STUDENT:

HÜSEYİN ERDOĞAN

040100054

# Introduction

Aim of this homework is learning how to write merge sort, insertion sort and linear search algorithms and decide which algorithms is better to use in what kind situations. First I read the "location.txt" file and according to value of N, I carry information to "sort s\_array ". After terminating sorting algorithms, program creates an output file named "location\_output.txt". In this file K cities are written.

## Development and Runtime environment

Project is developed in Microsoft Windows 7 . Microsoft Visual, Dev-C++ and g++ are used for compiling.

Program has two class. One of them is created to store info and other is created to sorting algorithms ,

In Point class:

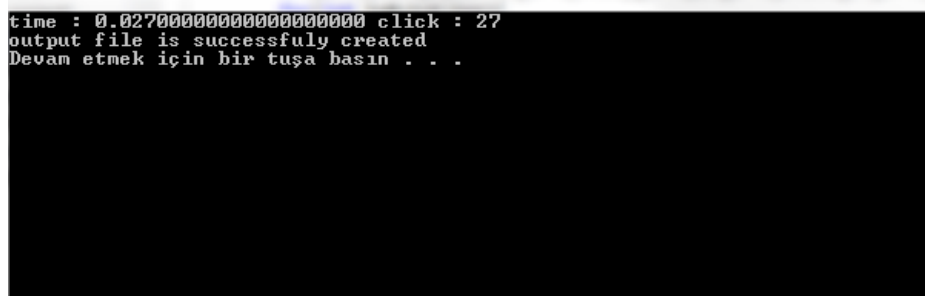
- **string city\_name** : for storing info about city's name
- **float latitude** : for storing info about city's latitude
- **float longitude** : for storing info about city's longitude
- **float distance** : for storing info about city's distance to reference point

In sort class:

- **Point \*city\_array** : for storing city array
- **int size\_of\_array** : size of city array
- **sort()** : for creating N sized array

- ~sort() : for deleting array
- void calculate\_distance(float, float) : for calculating distance to reference point
- void insertion\_sort() : for insertion sort
- Point\* merge\_sort(Point \*, int) : for merge sort and this function returns an array
- Point\* assistant\_merge(Point \*, int, Point \*, int) : assistant to merge\_sort function
- void linear\_search() : for linear search

When program is started, user enters to command line N, K, algorithm type, reference latitude and reference longitude . Program takes “location.txt” file and after execution creates “output\_location.txt” file. When program is terminated on the top time is showed and give information about output file.



```
time : 0.027000000000000000 click : 27
output file is successfully created
Devam etmek için bir tuşa basın . . .
```

In this lines I take info from command line.

```
int main(int argc, char* argv){

    int N = atoi(argv[1]);
    int K = atoi(argv[2]);
    float r_latitude = atof(argv[5]);
    float r_longitude= atof(argv[6]);
    string sort_type = argv[3];
```

User have to write like this format to command line :

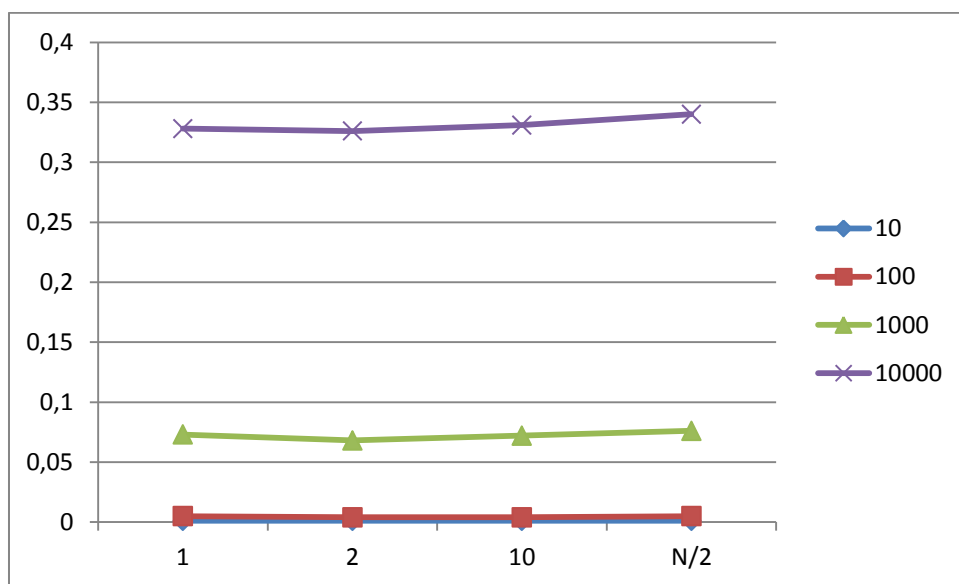
./studentID\_AoA1\_P1 N K algorithm type latitude longitude

./040100054\_AoA1\_P1 100 10 Merge Sort 0 0

# Algorithm Tables and Graphics

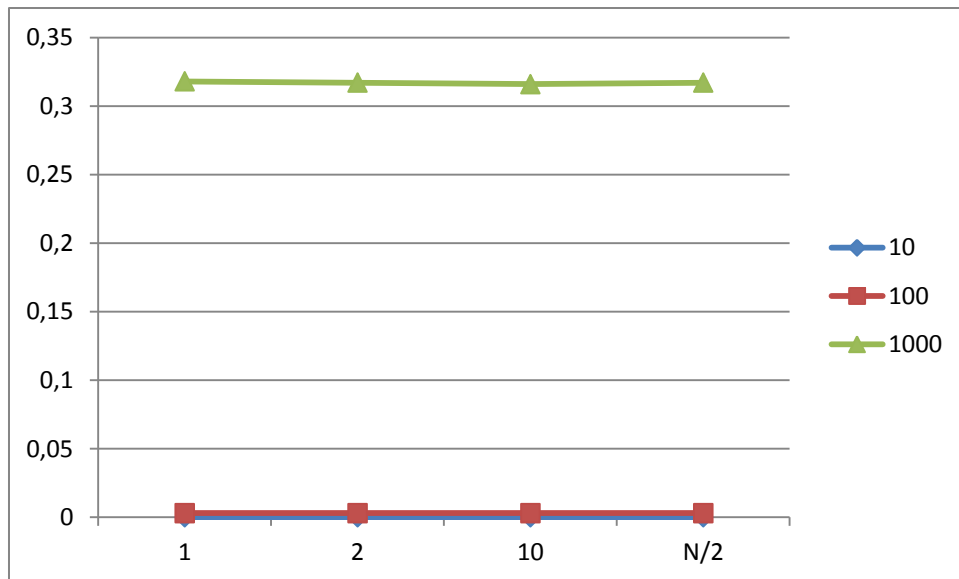
- Merge Sort

	1	2	10	N/2
10	0,001	0,001	0,001	0,001
100	0,005	0,004	0,004	0,005
1000	0,073	0,068	0,072	0,076
10000	0,328	0,326	0,331	0,34



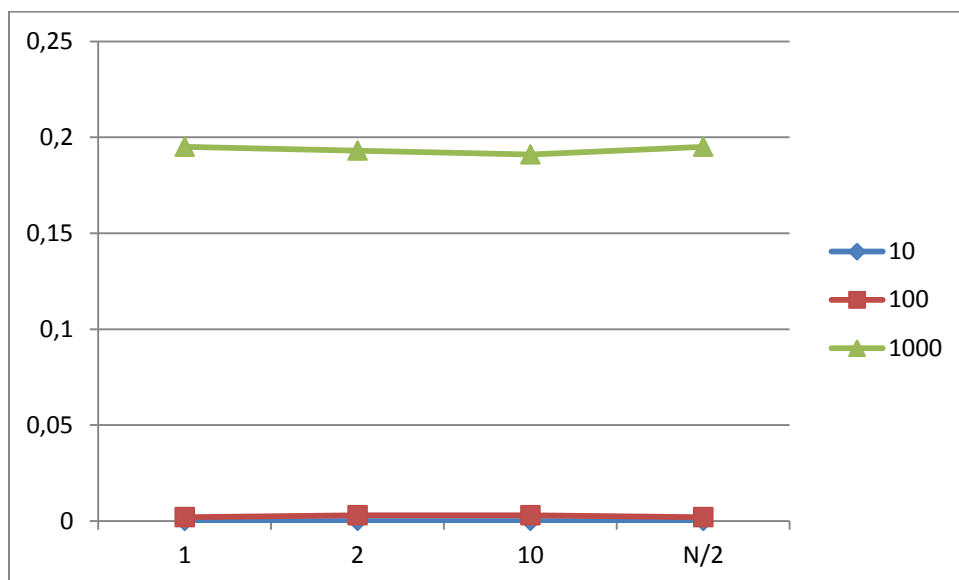
- Insertion Sort

	1	2	10	N/2
10	0	0	0	0
100	0,003	0,003	0,003	0,003
1000	0,318	0,317	0,316	0,317



- Linear Search

	1	2	10	N/2
10	0	0	0	0
100	0,002	0,003	0,003	0,002
1000	0,195	0,193	0,191	0,195



Result : I choose Linear Search if number is lower than 100. If number is bigger than 100, I choose Merge Sort.