

# Homework 1

## Submission Format:

Submit a zip named <Roll\_Number>.zip which on unzipping should have the following directory structure.

```
----<Roll_Number>
    |-----q1
    |-----q1.pdf
    |-----q2
    |-----q2.pdf
    |-----q3
    |-----<Roll_Number>_sc.py or <Roll_Number>_sc.cpp.
    |-----README.md
    |-----<Roll_Number>_report.pdf
```

Q1 -> Written question, no code required

Q2 -> Written question, no code required

Q3 -> Programming question: **your program must assume that the large dataset does not fit in main memory**, the report should contain your plots and observations, readme should contain instructions to run your program, and file format used (bin/txt).

## Execution and Input/Output format:

```
python <Roll_Number>_sc.py <input_file> <number_of_pivots>
./a.out <input_file> <number_of_pivots>
```

### (Text File Format)

Input format:

N, number of elements:  $1 \leq N < 1e10$

N will take up 10 positions (with leading zeroes, eg 1 will be represented as "0000000001", ie, **fixed width of 10**)

arr[i], array elements:  $0 \leq \text{arr}[i] < 1e6$

Array elements will take up 6 positions (with leading zeroes, eg. 1 will be represented as "000001", ie, **fixed width of 6**)

File format:

content: N followed by N lines (total N+1 lines): arr[i]<newline>

**N**  
**arr[0]**  
**arr[1]**  
....  
**arr[n-1]**

The first 10 positions represent the value of N, skipping the next position ('\\n'), the next 6 positions represent arr[0], skipping the next position ('\\n'), next 6 positions represent arr[1], and so on.

Output format:

sorted\_arr[i], array elements:  $0 \leq \text{sorted\_arr}[i] < 1e6$   
Array elements will take up 6 positions (with leading zeroes, eg. 1 will be represented as "000001", ie, **fixed width of 6**)

naming: <Roll\_Number>\_out.txt  
file format: N lines (sorted), N is the number of elements

**sorted\_arr[0]**  
**sorted\_arr[1]**  
....  
**sorted\_arr[n-1]**

The first 6 positions represent sorted\_arr[0], skipping the next position ('\\n'), next 6 positions represent sorted\_arr[1], and so on.

## (Binary File Format)

Input format:

N, number of elements:  $1 \leq N < 1e10$   
N will take up **8 bytes**.

arr[i], array elements:  $0 \leq \text{arr}[i] < 1e6$   
Array elements will take up **4 bytes**.

File format:

content: N followed by N numbers

**<N><arr[0]><arr[1]>....<arr[n-1]>**

The first 8 bytes represent the value of N, the next 4 bytes represent arr[0], the next 4 bytes represent arr[1], and so on.

Output format:

sorted\_arr[i], array elements:  $0 \leq \text{sorted\_arr}[i] < 1e6$

Array elements will take up **4 bytes**.

naming: <Roll\_Number>\_out.bin

file format: N numbers (sorted), N is the number of elements

**<sorted\_arr[0]><sorted\_arr[1]>....<sorted\_arr[n-1]>**

The first 4 bytes represent sorted\_arr[0], next 4 bytes represent sorted\_arr[1], and so on.