

INF236 - Assignment 1

Kristian Sørdal

March 2, 2024

Contents

1	Problem 1	3
----------	------------------	----------

1 Problem 1

The implementation of sequential radix sort has been implemented in the following manner.

Algorithm 1: Sequential Radix Sort

Input : n - The length of the array, b - Key size (how many bits to interpret as one digit)

Output: t - the time taken to sort the array

```

 $a \leftarrow$  array of random unsigned 64-bit integers of size  $n$ 
 $tmp \leftarrow$  partially sorted array of size  $n$ , initialized with 0.
 $buckets \leftarrow 2^b$ 
for  $i \leftarrow 0$  to 64 by  $b$  do
     $bucketSize \leftarrow$  array of size  $buckets$ , initialized with 0.
    for  $i \leftarrow 0$  to  $n - 1$  do
         $\_ bucketSize[(a[i] \gg shift) \& (buckets - 1)]++$ 
    transform  $bucketSize$  into prefix sum array
for  $i \leftarrow 0$  to  $n$  do
     $val \leftarrow a[i]$ 
     $bucketIndex \leftarrow (val \gg shift) \& (buckets - 1)$ 
     $tmp[bucketSize[bucketIndex]] \leftarrow val$ 
 $a \leftarrow tmp$ 
  
```

So in order to compute an equation $T(n, b)$ which yields the time the sequential program takes to run, we need to consider the amount of FLOPS the sequential program uses, and the clock speed of the CPU on Brake.