

# Parallel Homework 5

Christian Winwood  
A02074021

February 22 2021

## 1 Implementation

My implementation was fairly simple. I mainly just followed the steps to create a bitonic sort. I start off by calculating the number of loops that I have to do and have each process calculate a random number and print out their number. Afterwards I loop through the different steps that I have to do in order to make the sort work. In each step, the processes calculate who they need to swap with and swap with that process. They then determine if they are doing a descending or an ascending sort and choose the correct number accordingly, either their own, or the one swapped with the other process. They then keep this as their number. When the entire process has been finished, each process prints out their own number and the program ends.

## 2 Compile and Run

The compile and run commands are the same they have always been. First you use

```
mpic++ bitonic.cpp
```

in order to compile the program. This creates an a.out files that you need to run. To run it you use the command

```
mpirun -oversubscribe -np jnumber-of-processes; a.out
```

and the program will run.

## 3 Desired Output

The output will be fairly simple. There is a banner saying if the numbers are the starting or the ending numbers. Afterwards there is a list of each process and their number that they have.