CS610: Assignment 2

Vishal Deka
21111271

September 4, 2022

1 Problem 1

For the terms A (i , i - j) and A (i , i -j -1), $i = i + \Delta i$ which means $\Delta i = 0$. Again, $i - j = i + \Delta i - j - \Delta j - 1$. Putting the value of Δi, $\Delta j = 0$. So the distance vector in this case is (o,-1) which not a valid dependence since the leading non-zero element is not positive. Similarly, for the 2nd term A (i +1 , i - j), there is no valid dependence. In case of the term A (i -1 , i -j -1), we have $i = i + \Delta i - 1$. So, $\Delta i = 1$ and similarly $\Delta j = 1$. So the distance vector is (1,1). Thus there is a valid flow dependence.

2 Problem 2

To find the inversions there is a brute force algorithm that uses 2 nested loops leading to a squared time complexity as shown below.

```
for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (arr[i] > arr[j])
                inv_count++;
```

Here the outer loop can be executed in parallel since there are no loop carried dependencies in the outer loop. To do this, elements are fairly allocated to the threads and perform the inner loop operations as is in each thread. Later, add up the individual counts to get the total number of inversions.

3 Problem 3

Three mutex locks are required: one for the shared buffer to ensure mutual exclusion between producers and consumers trying to access the buffer at the same time and one lock for each file input.txt and prime.txt to ensure mutual exclusion among producer and consumer threads respectively.

4 Problem 4

When enqueue is called by the client program for the first time, an initialization function is called which spawns 4 threads which are the workers. There are 2 mutex locks - one for the job queue and another for the key value store. One issue is that for operations like delete and update the client may provide a non-existent key. Since these operations internally call the Find() operation, they should also be able to return some signal to the client program.