Homework 4

Mitchell Gant[1]

[1]Georgia Institute of Technology

March 27, 2019

Problem 1

Given P_i = the amount of elements in processor with rank i with a total of p processors in the range of $[0, (p-1)]$

$$P_i = \frac{n}{p} - \frac{n}{p^2}, \text{ where } i = 0$$
$$P_i = \frac{n}{p}, \text{ where } 0 < i < (p-1)$$
$$P_i = \frac{n}{p} + \frac{n}{p^2}, \text{ where } i = (p-1)$$

Problem 2

We can append the destination rank to the end of message m_{ij} and then run sort algorthm on the message using the destination rank for comparisons. This will lead to the message ending up in the correct processor. The runtime and communication time will remain the same as the sorting algorithm at $T_{comp}(n,p)$ and $T_{comm}(n,p)$.

Problem 3

Given that the we can assume our real topology contains $2^{\lceil \log p \rceil}$ processors. We can embed a virtual array topology into our hypercube typology by simply mapping the first $[0, p-1]$ processor ranks in our real topology to their corresponding graycodes for the virtual topology. We can then discard the use of the remaining processors we don't need.

Problem 4

Below are pseudocode functions for mapping ranks in the real topology of an array onto a virtual ring topology and the inverse function of mapping the ranks of the virtual ring topology to the real array topology.

```
//array to ring
func ring_rank(int array_rank, int num_proc){
    if (array rank == EVEN) {
        return (array_rank / 2)
```

```
    } else {
        return (num_proc - ((ring_rank / 2) + (array_rank % 2)))
    }
}


//ring to array
func array_rank(int ring_rank, int num_proc) {
    if (ring_rank <= ceil(num_proc / 2)) {
        return ring_rank * 2
    } else {
        return (2 * (num_proc - ring_rank) - 1)
    }
}
```

Problem 5

(a) 722

(b) (5, 11, 7)