CS 255 – Homework 3

1. Modify the Box Sort algorithm to make CREW PRAM algorithm that selects the median of *n* input numbers in $O(\log n)$ steps using $n\log n$ processors. Assume that the *n* input numbers are initially located in global memory locations 1 through *n*.

   Solution:

   Step1: Randomly select $\sqrt{n}$ elements and sort them using $\sqrt{n}$ processors ($\frac{n}{\log n} > \sqrt{n}$)
   - Quicksort Algorithm O (log $\sqrt[2]{n}$) which is O (log(n))

   Step2: Divide the array into sub array of size $\log\sqrt{n}$ and assign them to each of the $\frac{n}{\log n}$ processors. (Total elements = $\frac{n}{\log n} * \log\sqrt{n} = \frac{n}{2}$)

   Step3: Every processor inserts each of the $\log\sqrt{n}$ elements between the splitters

   O (log $\sqrt[2]{n}$) = O(logn)

   Step4: Repeat Step2 and Step 3 for remaining $\frac{n}{2}$

   Step5: We have inserted all n elements between $\sqrt{n}$ splitters. Check the element at index $\frac{n}{2}$, if it is one of the splitter element return it as the median else consider the box that contains value at index $\frac{n}{2}$, if the size is < log n sort the elements using log sort otherwise recurse and return the element at the index $\frac{n}{2}$ as the median

2. Give a concrete example with 8 processors where the faulty processor succeeds in foiling a threshold choice in the Byzantine agreement algorithm.

   Solution:

   No of processors n = 8; $P_1, P_2, \ldots, P_8$
   Maximum faulty processors $\frac{n}{8} = 1$; Say P_8 is faulty processor
   L = $\frac{5n}{8} + 1$; 6
   H = $\frac{6n}{8} + 1$; 7
   G = $\frac{7n}{8}$; 7

Processor	Vote	Faulty Processor Vote	Tally	Threshold	Decision
P_1	1	1	6	L = 6	1
P_2	1	1	6	L = 6	1
P_3	1	1	6	L = 6	1
P_4	1	0	5	H = 7	0
P_5	1	1	6	L = 6	1
P_6	0	0	5	L = 6	0
P_7	0	1	6	L = 6	1
P_8(faulty)	-	-	-	-	-

The faulty processor P_8 sends its vote as 0 to processors P_4, P_6 and 1 to all other processors making the tally to be less than threshold for processors P_4, P_6 and it to be at least equal to threshold for all other processors thereby preventing the good processors for a decision.

3. Carefully give a DMRC algorithm for determine which node in a communications network has the most incoming traffic. Assume the input consist of (key; value) pairs of the form $((i,j);n_{i,j})$ where the key is the pair (i,j) and $n_{i,j}$ is the number of bytes of traffic from node i to j in the network.

Solution:

Map Phase 1

map ((i, j); n_{ij})
    output (j; n_{ij})
return

In this map phase, we output keys as node with incoming traffic (j for pair (i, j)); trafficSize)

Reduce Phase 1

reduce (j; < n_{ij}, n_{kj} ... >)
    totalTraffic = 0;
    for v in < n_{ij}, n_{kj} ...>
        totalTraffic += v;
    output (j; totalTraffic)
return

In this reduce phase, we sum up all the incoming traffic for a given node

Map Phase 2

```
map (j; totalTraffic)
        output ("nodes"; (j, totalTraffic));
return
```

Map all keys values pairs to single key "nodes" and value as (node, traffic) pair

Reduce Phase 2

```
reduce ("nodes", < (i, totalTraffic), (j, totalTraffic), (k, totalTraffic) ……>)
        maxNode;
        maxTraffic = 0;
        for v in < (i, totalTraffic), (j, totalTraffic), (k, totalTraffic) ……>
                if (v. totalTraffic > maxTraffic)
                        maxNode = v. node;
                        maxTraffic = v. totalTraffic;

        output ("nodes", maxNode; maxTraffic);

return
```

In this reduce phase 2, we find the node which has maximum traffic

Modification to Parallel MIS to find Maximum Cliques

a) Construct the dual graph G^1 for given graph G

b) Run Parallel MIS algorithm on graph G^1

c) Return the output of MIS for graph G^1 algorithm as maximum clique for graph G

Why it works?

We construct the dual of graph G by adding edges for vertices pairs that do not have an edge in G and by removing edges for vertices pairs that have an edge in G.

An independent set is a set of vertices such that each vertex in set has no edge to any other vertex in the set. While a clique is a set of vertices with each vertex having an edge with every other vertex in the set.

So, finding MIS for dual graph G^1 is same as finding maximum clique for graph G.