

COS 226 – Data Structures and Algorithms
Fall 2014 – Flipped Lecture Section
Group Worksheet – 09.17.14
30 minutes

Instructions: Form a group of 3-4 to discuss each of these problems. Come up with a solution. If you have the right solution, you may be asked to come present it to the class. The key here is to pay close attention to specs. Runtime and memory constraints are critical to designing the proper data structure. Think of all data structures and algorithms you have seen so far and how if they apply (if any) to solving these problems. Answers will be provided later.

- 1. Social network connectivity.** Given a social network containing N members and a log file containing M timestamps at which times pairs of members formed friendships, design an algorithm to determine the earliest time at which all members are connected (i.e., every member is a friend of a friend of a friend ... of a friend). Assume that the log file is sorted by timestamp and that friendship is an equivalence relation. The running time of your algorithm should be $M \log N$ or better and use extra space proportional to N .

- 2. Bitonic max:** An array is *bitonic* if it consists of a strictly increasing sequence of keys immediately followed by a strictly decreasing sequence of keys. Design an algorithm that determines the maximum key in a bitonic array of size N in time proportional to $\log N$.

Example:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a[i]	10	34	56	76	87	80	70	66	56	30	28	25	20	15	11

3. Consider a datatype Successor with the following API:

```
public class Successor {  
    public Successor(int N) // creates a successor set of size N  
    public void remove(int x) // removes x if it exists in the set  
    public int successor(int x) // find smallest y in S such that  $y \geq x$   
    ...  
}
```

Design a data type so that all operations (except construction) take logarithmic time or better.

4. Describe a way to use a circular resizable array to implement a queue data structure where enqueue and deque takes amortized constant time.