Math 475 Exercises 3

Due: Apr 1, 2010

1. The procedure below takes an array of integers and determines if some elements occurs three (or more) times in the array. Which of the following big-O estimates: $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^2 \log n)$, $O(n^3)$, $O(n^3 \log n)$, $O(n^4)$, and $O(2^n)$ best describes the worst-case running time of the algorithm.

```
public boolean hasThreeEqual(int[] arr) {
  int n = arr.length;
  for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
      if (arr[i] == arr[j]) {
        for (int k = j + 1; k < n; k++) {
          if (arr[j] == arr[k]) {
            return true;
          }
        }
      }
    }
  }
  return false;
}
```

2. Describe (you don't need to write a program for it) a fast algorithm to do the same proceedure as problem 1.

3. Let V be a vector of size n representing a partition as described in class. Describe an algorithm that modifies V so that each tree (block) has depth at most 1 and runs in time $O(n)$.

4. Let V be a vector of size n representing a partition and suppose the trees have depth at most 1. Find an algorithm that runs in time $O(n)$ that changes V so that the root of each tree is the smallest member of that block (and the depth stays at most 1).

5. The lattice of all subsets of an n-element set has order dimension n; that is, it is the intersection of n linear extensions and no fewer. We outlined in class that it is the intersection of n linear extensions, so you don't need to do that part. But prove that it cannot be written as the intersection of fewer than n linear extensions.

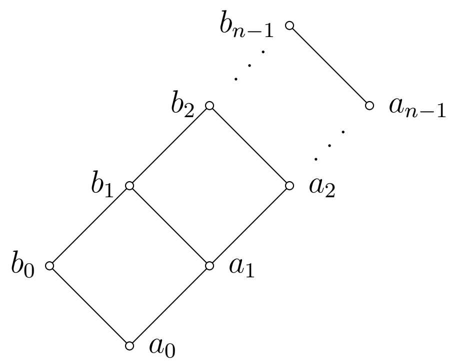

Figure 1: $\mathbf{2} \times \mathbf{n}$

6. Show the lattice in the picture has

$$\frac{1}{n+1}\binom{2n}{n}$$

linear extensions. You can use Theorem 8.1.1 in the book.