11:11 AM 10/4/2015

Sunday Meeting (11AM - 2 PM)

1) **Live Coding**

    * WAP: Given two arrays of numbers >= 0, find elements that are in array 1 but not in array 2.

        Ex) arr1: 1,3,7,0,5,12;
            arr2: 1,2,4,7,9,3;
            output: 0,5,12

    * int main()

```c
int main()
{
        int arr1[] = {1,3,7,0,5,12};
        int arr2[] = {1,2,4,7,9,3};
        int i = 0, j = 0;
        int size = 6;
        int found = 0;
        for (i = 0; i < size; ++i)
        {
                for(j = 0; j < size; ++j)
                {
                        if(arr1[i] == arr2[j])
                                found = 1;
                                break;
                }
                if(!found)
                        printf("%d", arr1[i]);
                found = 0;
        }
        return 0;
}
```

```
// That was O(N^2) complexity
* // Let's try to make it less complex
    // Use a look-up table and set values to on or off. Ex) 0 1 2 3 4 5 6 7 8 9 10 11 12
 //                                                        [0 1 1 1 1 0 0 1 0 1   0   0   0]


        int main()
        {
                int size = 6;
                int size_m = 12;
                int lookUp[size_m + 1];
                int i = 0, j = 0;
                for(i = 0; i < size_m + 1; ++i)
                {
                        lookUp[i] = 0;
                }
                for (i = 0; i < size; ++i)
                {
                        lookUp[arr2[i]] = 1;
                }
                for
                {
                        if(!lookUp[arr1[i]])
                                printf("%d", arr1[i]);
                }
        }
    // What's better about the second program is that the complexity is O(N), the worst case is that
there are 13 iterations instead of 36.
        // What's worse is that there is a third array that we implement, which means more memory.
    // The first program/solution is a naive solution, the second one is better.
```

2) **GitHub**

   * Basic steps: make changes, add changes, commit, push
   * You can download github and then Git Shell, or do it through Cygwin.
   * git init //makes a GitHub directory
   * git remote add <name> <URL>
   * git remote -v
   * git pull <name> master
   * make a file
   * git add <file-name>
   * git commit -m "message"
   * git push <name>
   * Remove file from your local directory: rm <file-name>
   * Commit the changes: git commit -m "message"
   * Push the changes: git push <name> master
   * * CANNOT COMMIT EMPTY FILES! *
   * to save/update everything: git fetch <name>; git rebase <name>/master

3) **Ilya flashed code onto the Quad.**