The Stony Brook Computing Society

April 6, 2016

Microsoft Tour

- Friday, April 15th
- Tour the Times Square office and attend a Q&A session!
- Signups are now closed

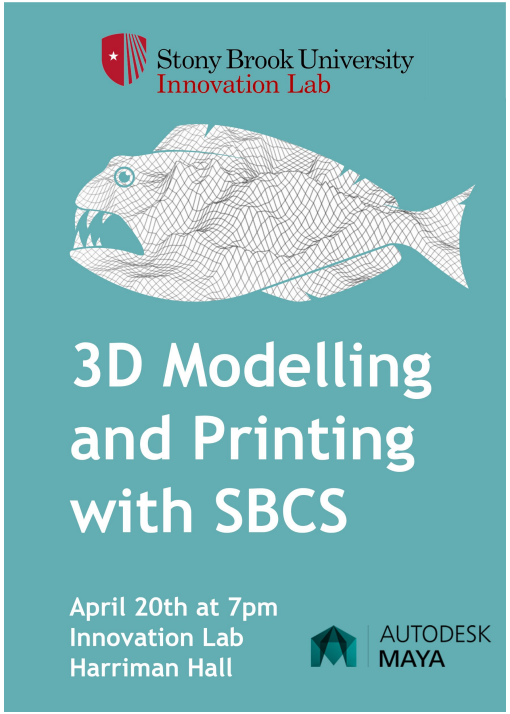
SBCS Workshop: 3D Modeling & 3D Printing

What: 3D modeling using Autodesk Maya, and submit your creations to the innovation lab to be printed!

When: Wednesday April 20th, 7:00pm

Where: Innovation Lab

Prerequisite: please download and install Autodesk Maya – Click Here to Download



The Stony Brook Computing Society

SBCS Workshop: UI/UX & Graphic Design

- Different day!!!
- When: Thursday, April 28
- Learn basic UI/UX with Phil!
- Free pizza 🍕🍕🍕

Elections!

Nominations are up: http://goo.gl/forms/FUwlvhQsrE

Elections will take place on April 13th!

Practice Interview Question

1. Let's say you have a list of N+1 integers between 1 and N. You know there's at least one duplicate, but there might be more. For example, if N=3, your list might be 3, 1, 1, 3 or it might be 1, 3, 2, 2. Print out a number that appears in the list more than once.
2. Now you want something faster than O(n^2), you can only use constant space, and you can't manipulate the original list.

Solution

Go through the list and count the number of integers between 1 and N/2. If the count is greater than the number of possible integers in that range, then I know there's a duplicate in that range. Otherwise, a duplicate must exist in the range of N/2+1 to N. Once I know which half of the range the duplicate is in, I can recurse and binary search in that half, then keep repeating the process until I've found a duplicated number. The time complexity is O(n*log n) and the space complexity is O(1).

As always, feel free to contact us!

- Website: http://sbcs.io/
- Email: eboard@sbcs.io
- Facebook group: https://www.facebook.com/groups/sb.computing/
- Slack team: https://seawolf.slack.com/signup