**Weekly Course Schedule**

***9/19: Week 1: Syntax/Java Review***

-Self introductions/icebreaker (what language - C++/Java - you prefer to work with, your CS background, grade, name, etc)

**-Introduce USACO exam format and similar contests**

-**Create an USACO and Springlight Online Judge Account (if haven’t done so)**

**-**Review (or learn?) basic coding concepts

-Slides: Week 1 slides <https://docs.google.com/presentation/d/1s3Y_h1ANSGAUTvajRg89A4NmFnE4my0PdAnK0UfyEpA/edit#slide=id.p1>

**-Homework**: Finish and understand basic Java exercises (1..100 tower, for loops, if statements, basic Java/C++ syntax).

-Contest Timing

-Haybales

***9/26: Week 2: String Operations and Intro to Brute Force/Runtime:***

Slides:

Week 1 slides <https://docs.google.com/presentation/d/1s3Y_h1ANSGAUTvajRg89A4NmFnE4my0PdAnK0UfyEpA/edit#slide=id.p1>,

Week 2 and 3 <https://docs.google.com/presentation/d/1ZLVvhbCD0_ChtEOi5J1aO7brU1roe-qXw9_BKP9mdFg/edit#slide=id.p7>

**-Review** week 1 problems

**-**Intro to runtime and Big-Oh notation/analysis - 10^8 operations is around 1 second, is a little expensive time-wise. O(N) means the program takes N operations on average; O(N^2) means the program takes N^2 operations on average.

-Brief discussion about brute force and significance in Bronze

- String problems - MooCast (solve at home - optional)

- **Homework:**

**Implement Why Did the Cow Cross the Road:** <http://www.usaco.org/index.php?page=viewproblem2&cpid=712>

Begin Photoshoot: <http://www.usaco.org/index.php?page=viewproblem2&cpid=988>

Begin Block Game: <http://www.usaco.org/index.php?page=viewproblem2&cpid=664>

Week 3: Photoshoot/BlockGame/WhyDidTheCowCrossTheRoad review/ Brute Force intro/brief primer on Two Pointers

* Week 3 slides: <https://docs.google.com/presentation/d/1ZLVvhbCD0_ChtEOi5J1aO7brU1roe-qXw9_BKP9mdFg/edit#slide=id.p7>
* Reviewed Why Did the Cow Cross the Road (check <https://docs.google.com/document/d/1a4iPiyq98az8MxYGB56IbAjKVnn50gfrggjpKzYL46g/edit> )
* Reviewed Photoshoot/BlockGame
* Covered brute force/DiamondCollector and BovineGenomics
* Brief intro to two pointers way of solving DiamondCollector
* **Homework:** Finish/fix previous problems (Why Did the Cow Cross the Road), Photoshoot, Block Game
* **Start** Diamond Collector, Bovine Genomics, and Hoof Paper Scissors (see Week 3 slides for problems)

1 6 4 3 1

**1** **1 3 4** 6

K = 3

***1* 1 2 2 3 3 3 3 3 3 4 *4*** 5 6 6 6 7 7 7 8 9 10

1 1 ***2* 2 3 3 3 3 3 3 4 4 *5*** 6 6 6 7 7 7 8 9 10

1 1 2 2 **3 3 3 3 3 3 4 4 5** **6 6** 6 7 7 7 8 9 10

1 1 2 23 3 3 3 3 3 **4 4 5** **6 6** **6 7 7 7** 8 9 10

op to check maximum number of diamonds, 2N operations to move left&right pointer

O(N log N) - sorting

O(2N) to move O(1) -

Week 4: Review previous problems/Intro to 1D problems/a little bit of PythonChallenge

* Covered review problems
* The Lost Cow - <http://www.usaco.org/index.php?page=viewproblem2&cpid=735>
* Why did the Cow Cross the Road III - <http://www.usaco.org/index.php?page=viewproblem2&cpid=713>
* HW: Fix all previous problems - if any issues, ask
* HW: try out Mixing Milk; <http://www.usaco.org/index.php?page=viewproblem2&cpid=855>
* HW-optional Hard - Bessie Gets Even (brute force) <http://www.usaco.org/index.php?page=viewproblem2&cpid=546>
* HW-optional Hard - Angry Cows (1d simulation) <http://www.usaco.org/index.php?page=viewproblem2&cpid=592>

- direction: positive/negative

- number of zigzags(?)

- number of moves

- heading: where we are heading (2^(zigzags-1))

- current location (initially x)

direction = +1

moves = 0

heading = 1

curr\_loc = x

while(curr\_loc != y) {

curr\_loc += direction;

moves++

if(curr\_loc == x + heading\*direction) {

direction \*= -1;

heading \*= 2;

}

}

print(moves);

10/17 Week 5 - Intro to stacks/queues (not directly covered in Bronze), Angry Cows Bronze

* **1st Mock Contest:** [**http://www.usaco.org/index.php?page=dec19results**](http://www.usaco.org/index.php?page=dec19results) **(priority 1 - IMPORTANT. Time yourself and do the contest for 4 straight hours, it’s an “open book” contest with all websites allowed (no human collaboration). Please come prepared with questions to ask me to make sure you know what you did wrong/could improve on.)**
* Try out Angry Cows Bronze - <http://www.usaco.org/index.php?page=viewproblem2&cpid=592> (priority 2 - do if only you’ve finished 1st mock contest. We will also discuss next week if needed.)
* Talked about stacks/queues
  + Reverse Polish Notation **(optional)** <http://138.197.210.211/JudgeOnline/problem.php?id=1074>
  + Parenthesis **(optional):** <http://138.197.210.211/JudgeOnline/problem.php?id=1073> (low priority)

Read in K, N

rank[K practice sessions][cow]

For (int i = 0; i < K; i++) {

For (int j =1; j <= N; j++) {

X <- Read in j-th number at the i-th row

Rank[i][x] = j

}

}

For (int a = 1; a < N; a++) {

For (int b = a+1; b <= N; b++) {

Boolean a\_before\_b = true;

Boolean b\_before\_a = true;

For (int k = 0; k < K; k++) {

if(rank[k][a] < rank[k][b]) b\_before\_a = false;

if(rank[k][b] < rank[k][a]) a\_before\_b = false;

}

if(a\_before\_b || b\_before\_a) {

consistent++

}

}

}

print(consistent)

<https://repl.it/repls/YummyPoliticalPresses>

**Week 6: 10/24 Review previous week practice test, discussed other contests:**

**Practice Test (4hrs)**: <http://www.usaco.org/index.php?page=feb20results>

Make sure you guys correct last week’s problems given the above psuedocode/approach.

If time remains, try out Angry Cows (Bronze) and the Stack stuff (Polish notation/Parentheses).

**Week 7: 10/31 Do the December 2017 Bronze Contest**

-Discussed previous week’s problems. Code for Q2: <https://repl.it/repls/AdolescentAffectionateConcentrate>

Practice Test (4hrs): <http://www.usaco.org/index.php?page=dec17results>

**Week 8: 11/7 Do the US Open Contest 2019, Talked a little about binary search:**

<http://www.usaco.org/index.php?page=open19results>

**Week 9: 11/14: Do January 2018 USACO Bronze, talked a little about DFS/Dijkstra/Floyd-Warshall and Recursion**

[**http://www.usaco.org/index.php?page=jan18results**](http://www.usaco.org/index.php?page=jan18results)