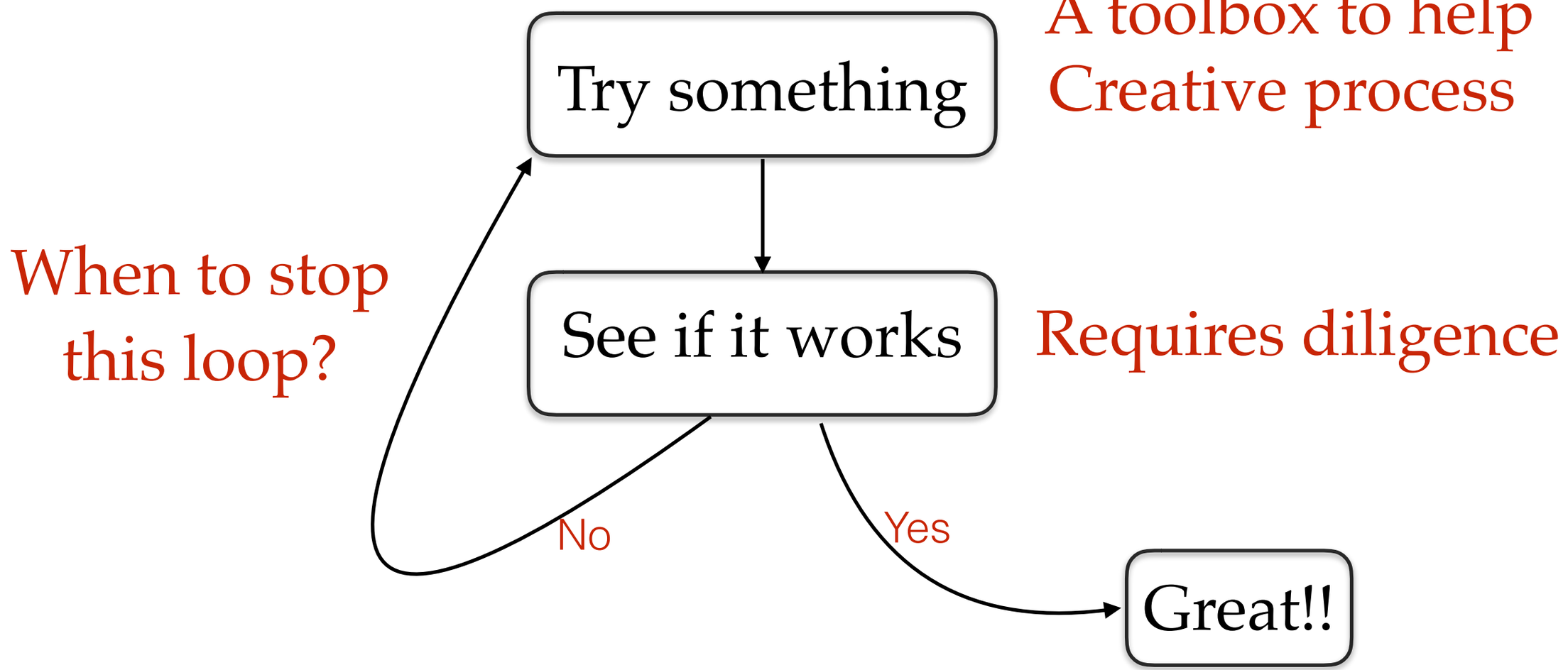
CS601
Algorithms and Complexity

Instructor: Rohit Gurjar
TAs : Abhinandan Singh, Jalay Shah, Nilesh Tanwar,
Vivek Kumar, Roshan Raj

https://www.cse.iitb.ac.in/~rgurjar/CS601/

Objectives

- How to design algorithms.



Algorithm's correctness

- How to argue that an algorithm is **not** correct: show bad examples
- Arguing algorithm's correctness
  - **Correctness by confidence**: I came up with it, so it must be correct
  - **Correctness by examples**: because it works for my favorite examples
  - **Correctness by authority**: It's just obvious

- Formal proofs of correctness
  - if it is not obvious to the CS601 class

Pre-requisites

- Not recommended for students who have done or will do CS218/CS218M.
- **Prerequisites**:
  - Comfortable with thinking at abstract level
  - Basic run-time analysis, O-notation, Recursion.
  - Basic data structures -- array, list, tree.
  - Basic probability and combinatorics.
  - Basic graph theory -- cycles, trees, depth-first search, breadth-first search.
- Check the self-assessment quiz (course webpage)

What will you learn

✦ **Principles of designing and analyzing algorithms:**

- Basic principles like induction/recursion.
- Divide and Conquer,
- Dynamic Programming,
- Greedy Algorithms.

✦**Beyond the basics:**

- Bipartite Matching
- Network Flow.
- Reductions.

✦**Complexity:**

- Polynomial time and the Complexity classes NP, co-NP.
- NP-hardness and NP-completeness.

✦**Randomization in Algorithm Design:**

- Random variables,
- Linearity of expectation,
- Applications: Approximate Max-cut, Min-cut, Quicksort. Hashing.

How will we do it

- Live lectures on Webex
- Recording will be uploaded on the course webpage.
- Communication via Moodle?
- Any suggestions are welcome.

Grading (tentative)

- Weekly Moodle quizzes (best n-2 out of n) - 10%
- 2 Programming Assignments - 20%
- 2 Theory assignments - 20%
- 2 or 3 Exams - (12.5+12.5+25) %
- Homeworks - 0%

References

- Kleinberg, Tardos (amazing book, freely available)
- Prof. Sundar's course notes
- Motwani, Raghavan: Randomized Algorithms