Important of Program Testing

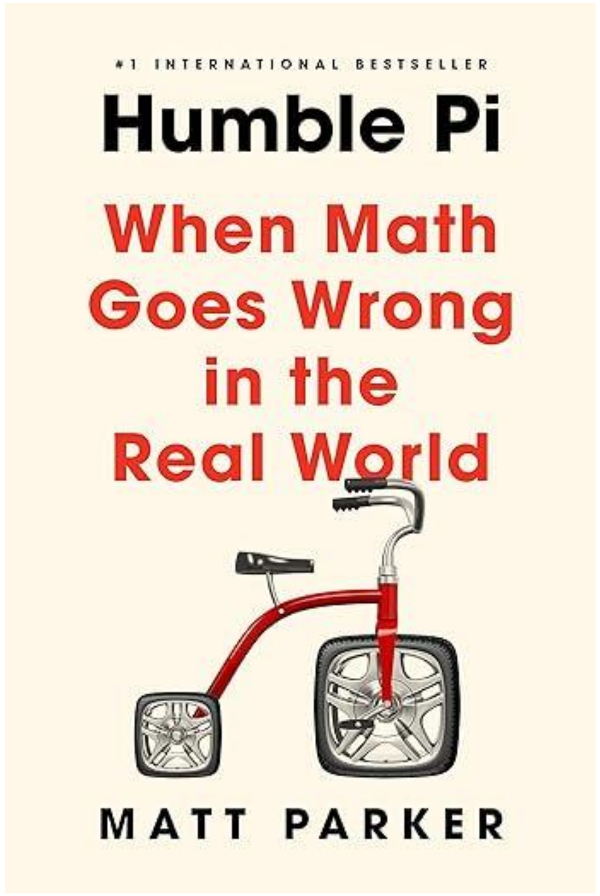
Arup Guha

COP 3502 – Computer Science I

11/30/2023

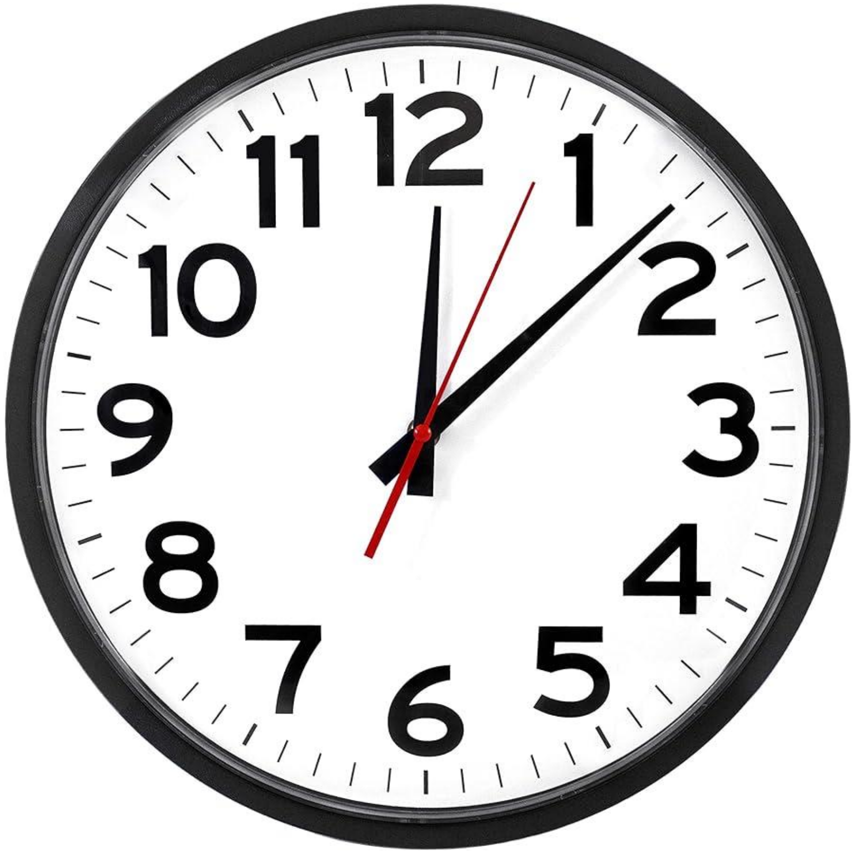
Book Summary: Humble Pi (Matt Parker)

- Most of the Math Discussed in this book comes down to poor testing/design by computer scientists or software engineers
- While there are domains of software engineering that require minimal math or testing, this book is a sober reminder that all code has to describe actions in a quantitative and unambiguous fashion.


#1 INTERNATIONAL BESTSELLER
Humble Pi
When Math Goes Wrong in the Real World
MATT PARKER

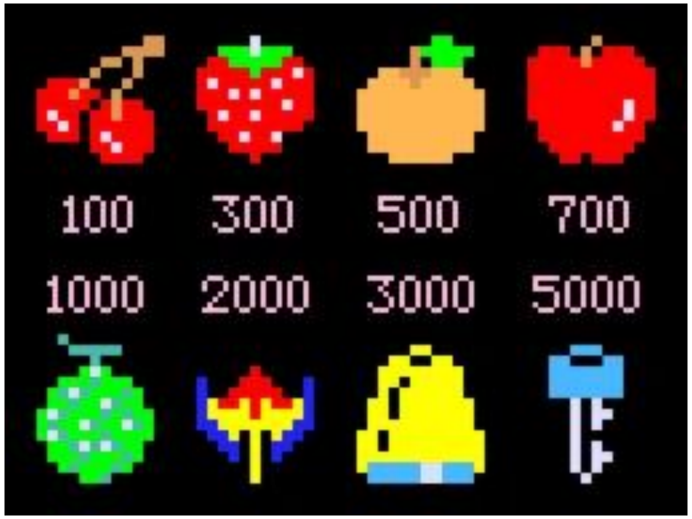
System Time

- Many programs use system time.
- 9/14/2004 – Southern California Air Traffic Control (49 days)
- Unix will overflow int sometime in 2038
- February 2007 – F-22 fighters (International Date Line)



Software Overflow Errors

- Therac-25 Shield (not main error)
- Civilization – aggressive Gandhi
- Pacman Fruit

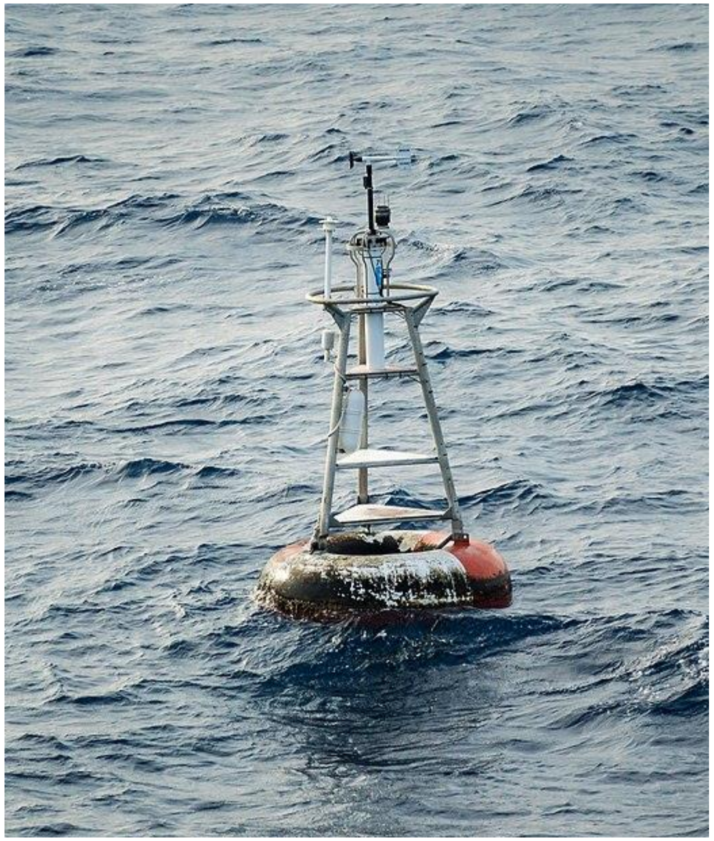


Software Imprecision Errors

- 2/25/1991 - Patriot Missile System (28 deaths)
- 1997 USS Yorktown (lost power due to division by 0)

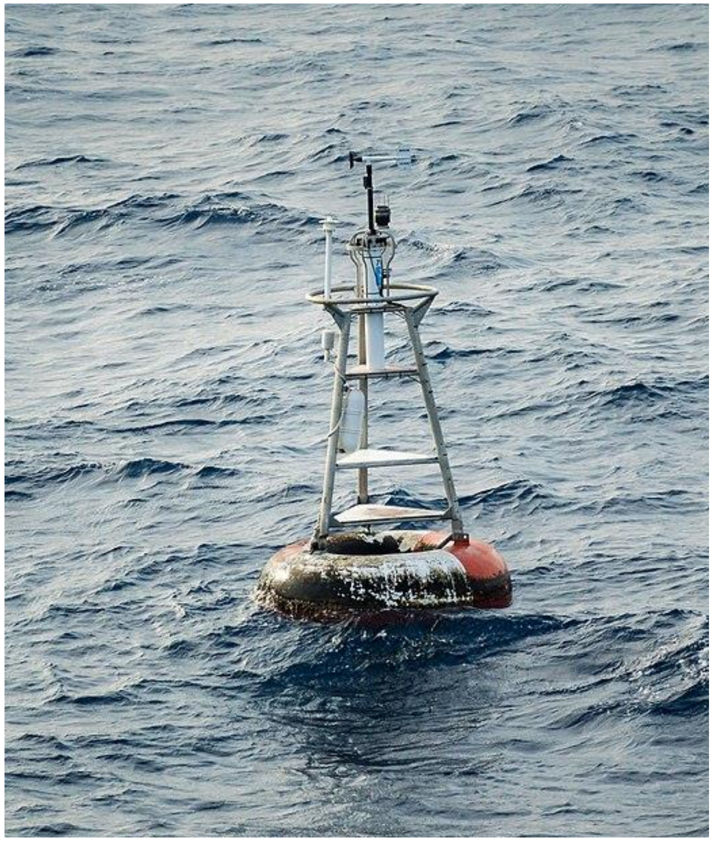
NULL Island

- Unknown crime locations in LAPD database stored as NULL.
- Steve Null (Sun employee)



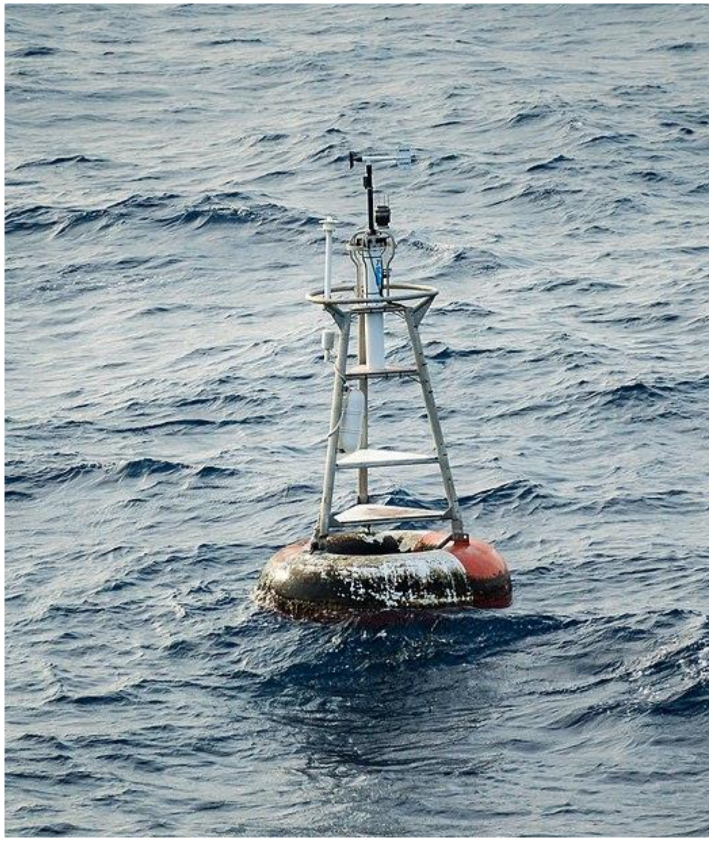
Spreadsheet Shenanigans

- Auto-number formatting
- Auto-date formatting (enzyme MARCH5)
- Max Rows (Wikileaks data missing)



Main Culprit: Logic Errors

- 2012 J.P. Morgan Chase (Value At Risk Error sum vs. avg)
- 1980 Texaco Oil Exploration Error
- Space Shuttle SRBs – measuring circularity
- 2012 J.P. Morgan Chase (Value At Risk Error sum vs. avg)



Conclusion

- While the grading criteria and test cases we make seem “mean”, they are nothing compared to the difficulty of real-life code or situations.
- Errors in real-life code can have serious consequences. It usually takes failure at several levels before these errors cause harm, but everyone in the chain should feel some level of responsibility.
- Details matter.
- One reason testing is difficult is because many situations are difficult to recreate in testing scenarios.