



# ON USING TESTING-RELATED TASKS IN THE IOI

Ahto Truu, [ahto.truu@ut.ee](mailto:ahto.truu@ut.ee)

Heno Ivanov, [heno@siil.net](mailto:heno@siil.net)

Estonia

# CONTENTS

- Introduction
- Scoring
- Case Studies
  - Task IF-THEN-ELSE
  - Task RELAY
  - Task QUICKSORT
- Conclusions



# INTRODUCTION

- Attributes
  - Black vs white box
  - Static vs dynamic
- Taxonomy
  - Static black box – the “problem-setter” mode
  - Dynamic black box – the “bug-compatible” case
  - Static white box – the “program analysis” case
  - Dynamic white box – the “debugging” case



# SCORING

- Goals
  - Correctness
  - Completeness
- Grading
  - Correctness: well-formed / valid / correct
  - Not correct: reject / ignore / penalize
  - Scoring: several (flawed) solutions
  - Scoring: per test case / failed solution / failure pattern
  - Scoring: coverage?
- Resources
  - K solutions per contestant

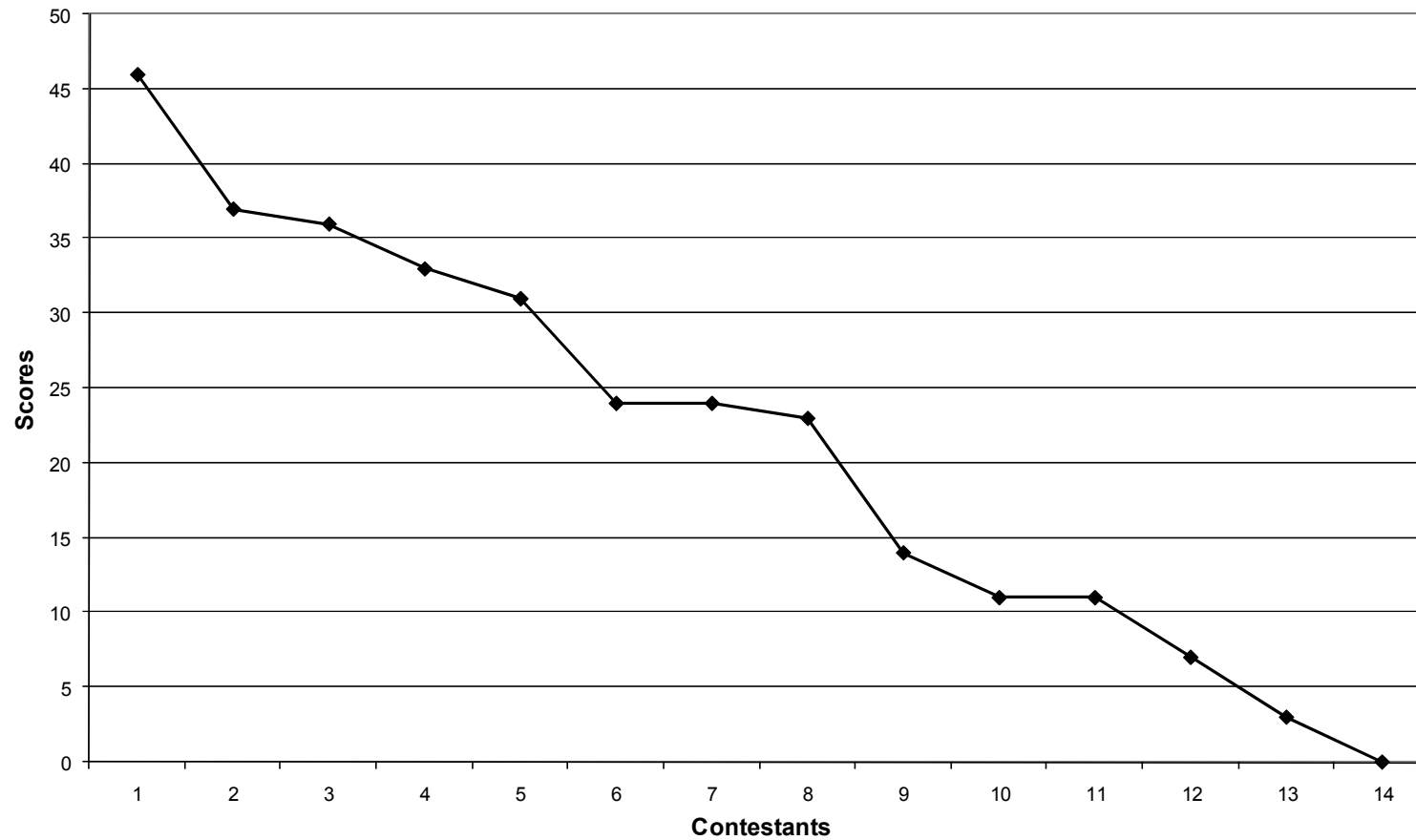


## TASK IF-THEN-ELSE (1)

- A very simple programming language
- Write an analysis tool to report all possible outcomes of a program given as input
- Classification:
  - Static white box
- Grading:
  - 11 independent test cases
  - 50 points total



## TASK IF-THEN-ELSE (2)

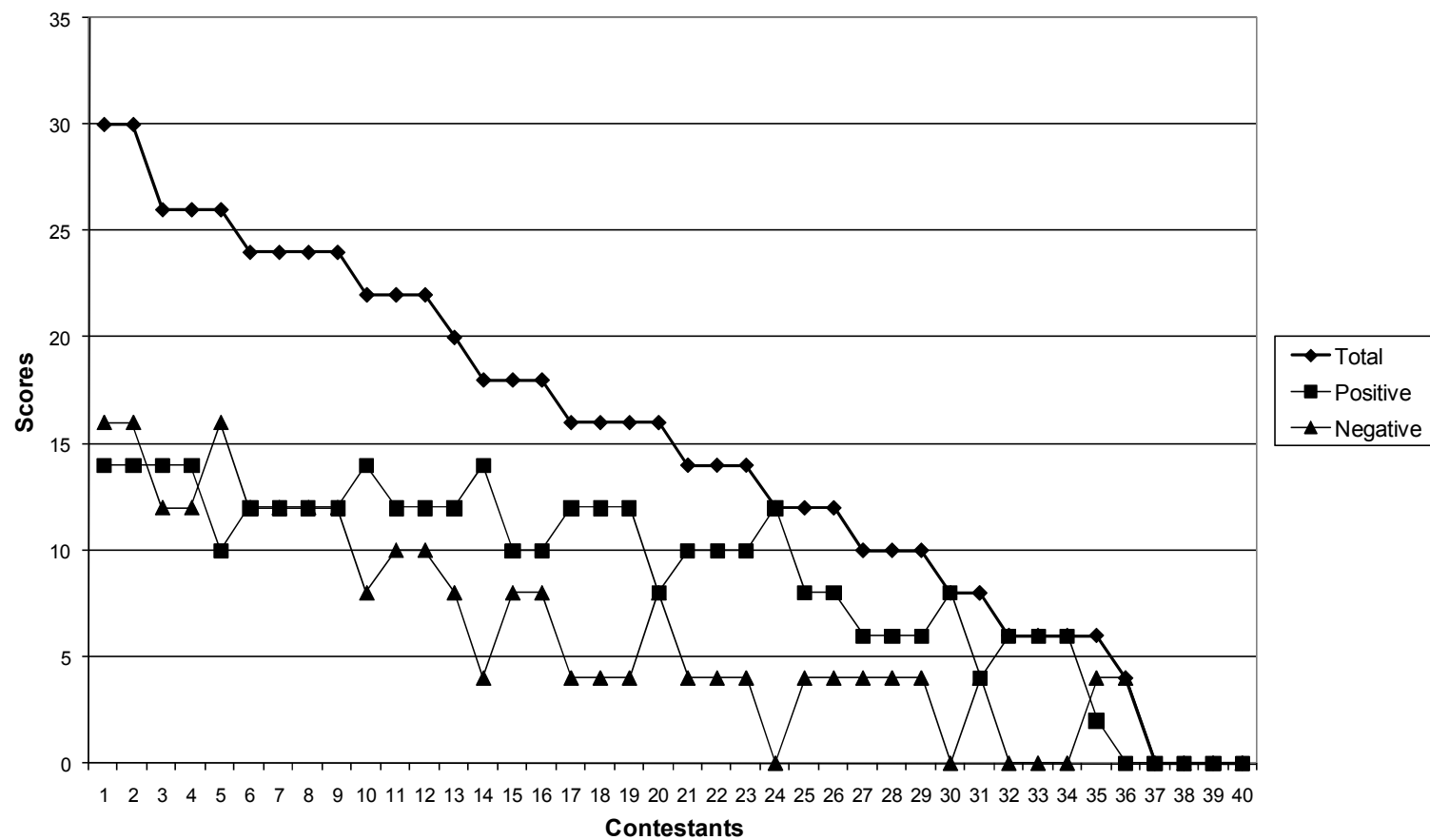


## TASK RELAY (1)

- A data filtering and sorting problem
- Binary implementation given
- Write a bug-compatible version
- Classification:
  - Dynamic black box
- Grading:
  - 7 “positive” test cases (2 points each)
  - 4 “negative” test cases (4 or 2 points each)
  - 30 points total



## TASK RELAY (2)

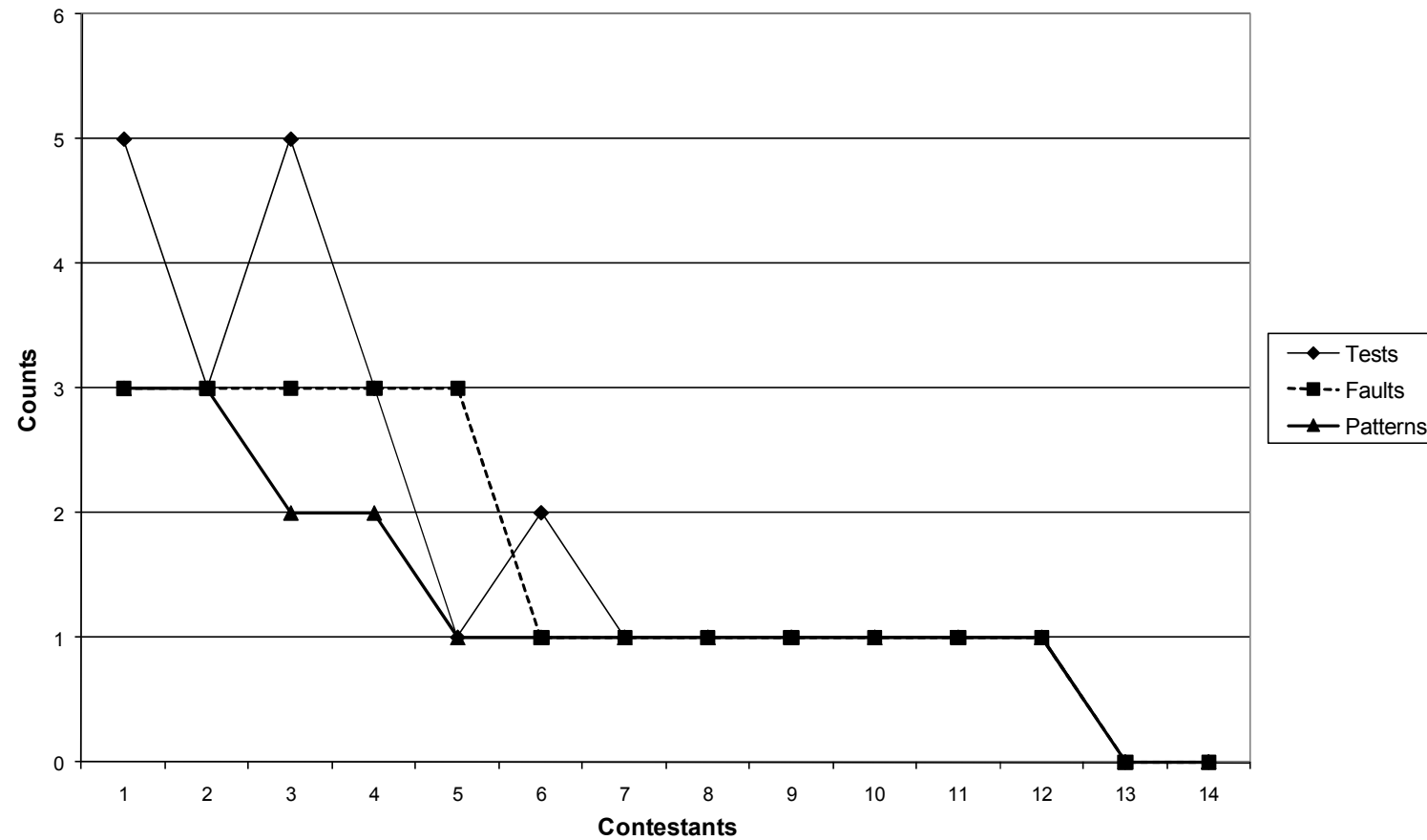


## TASK QUICKSORT (1)

- Several implementations of QuickSort
  - One correct, four have “<” vs “<=“ errors
- Create a set of test cases to recognize each implementation
- Classification:
  - Dynamic white box
- Grading:
  - Number of distinct failure patterns



## TASK QUICKSORT (2)



## CONCLUSIONS

- It is possible to create good testing tasks
- It does take effort to get it right
  - But this holds for other task types as well!



# THANK YOU

- Questions? Comments?

