

Structure, Scoring and Purpose of Computing Competitions (Part 2)

Gord Cormack

Graeme Kemkes

Ian Munro

Troy Vasiga (Presenter)

University of Waterloo

January 24, 2006

Introduction



- Yesterday, we discussed the landscape of (computing) competitions.
- This presentation discusses several directions of travel in this landscape.

Landscape Summary



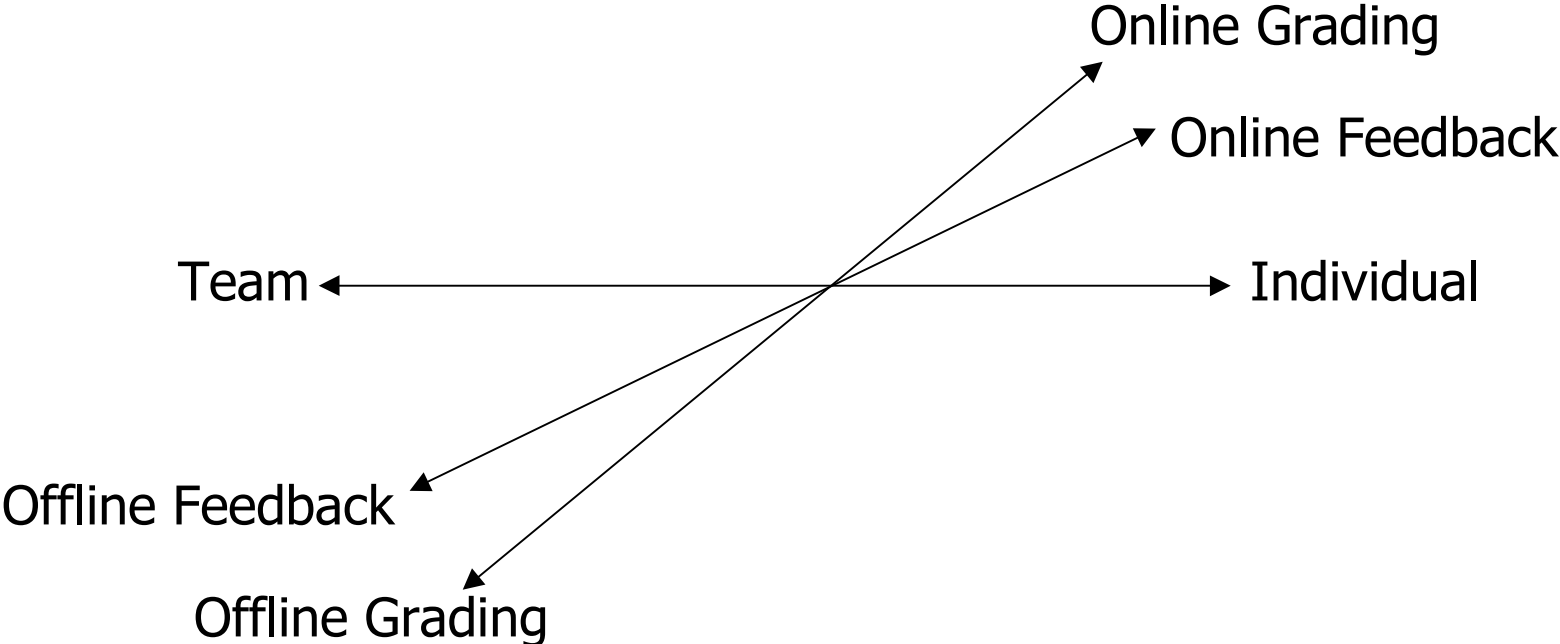
- team vs. individual
- on-line feedback vs. off-line feedback
- on-line grading vs. off-line grading
- restricted language set vs. open language set
- algorithmic focus vs. non-algorithmic focus
- computer-based vs. non-computer based
- projects vs. tasks
- long time vs. short time
- hardware vs. software vs. answer artifact

Landscape Map

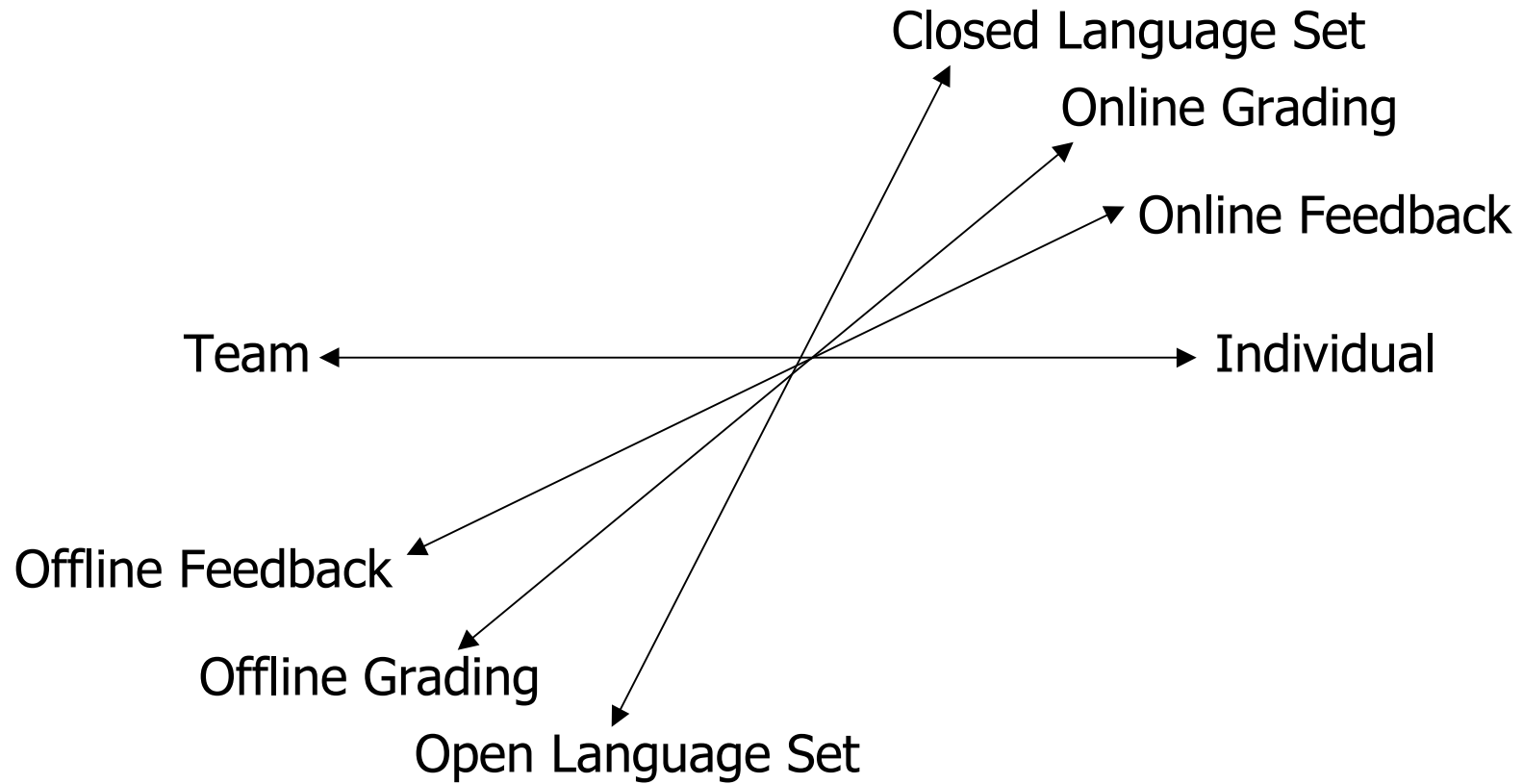


Team ←————→ Individual

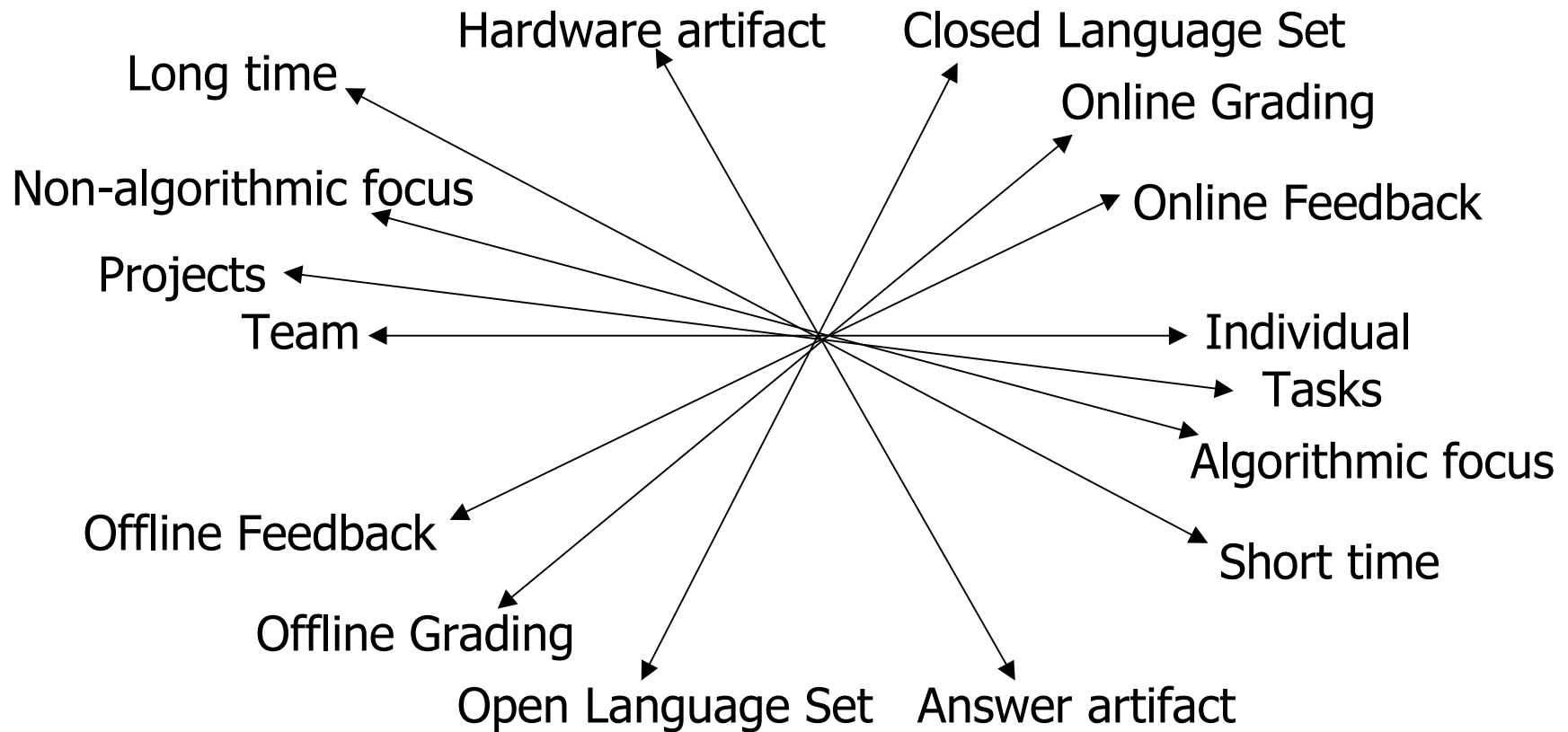
Landscape Map



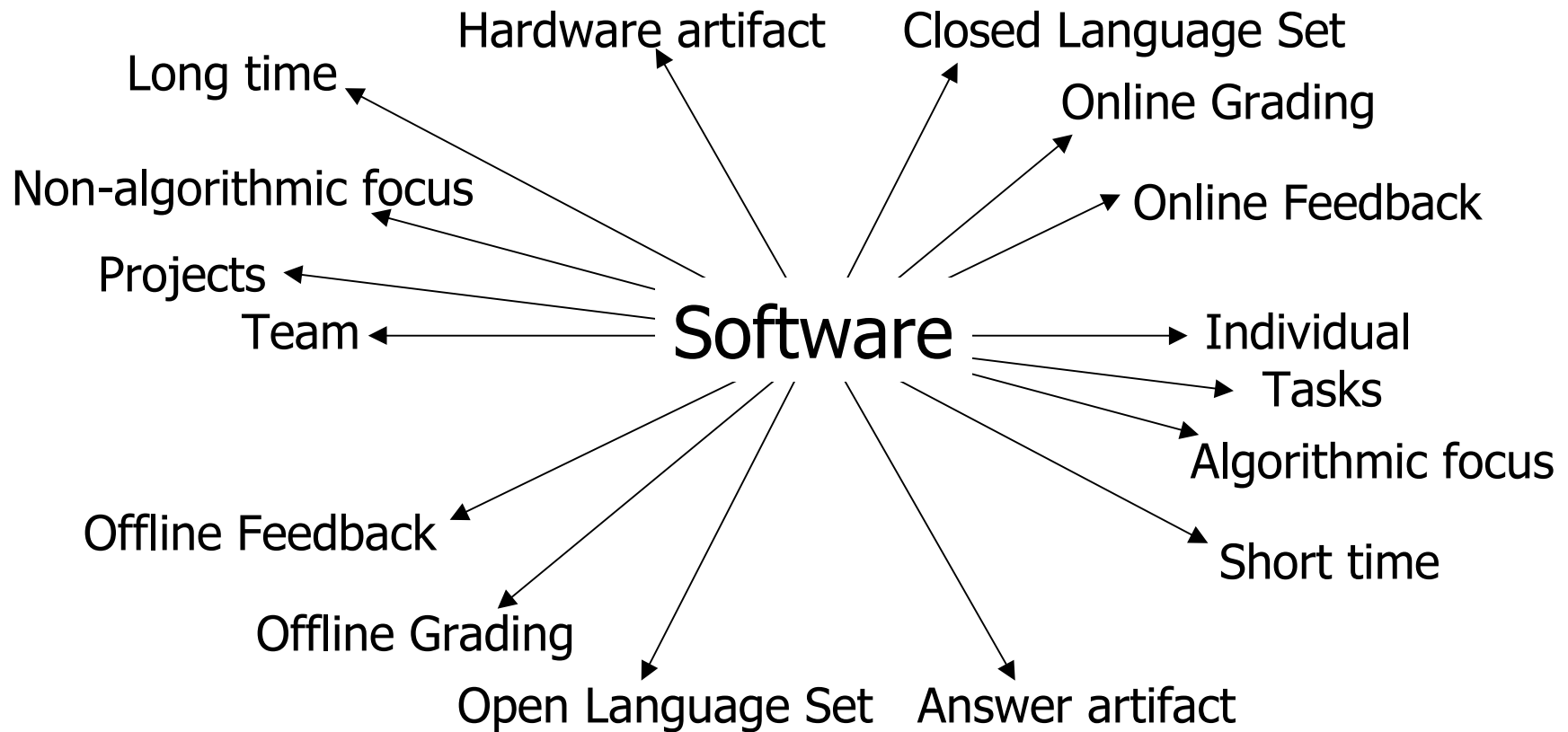
Landscape Map



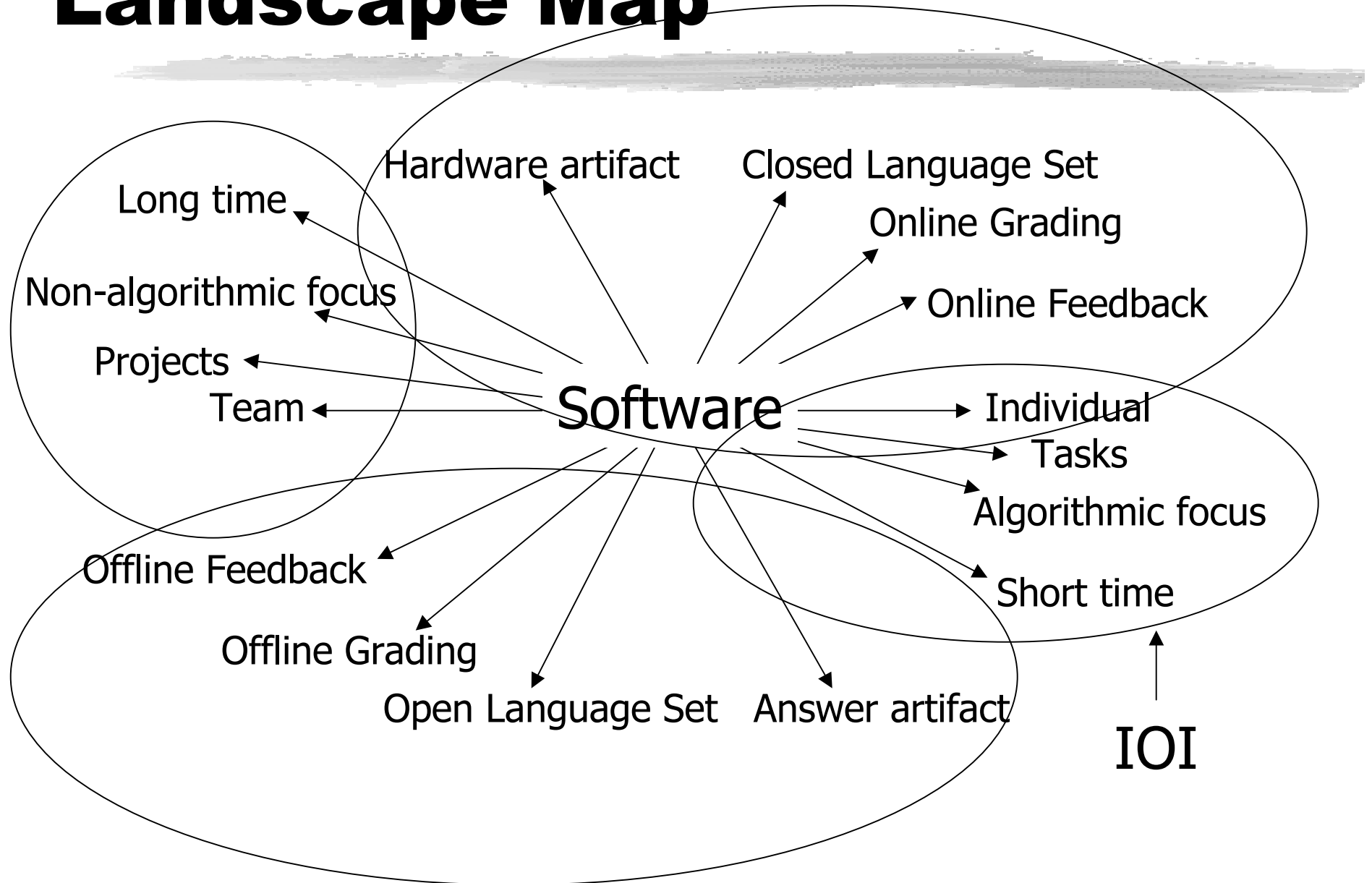
Landscape Map



Landscape Map



Landscape Map



How this affects us



- We will consider various proposals that attempt to improve/enhance one or more of these groups
- Each different contest can be placed somewhere along these dimensions

Specific Proposals



- Real-time feedback
- Test case creation task
- Graduated difficulty of tasks
- Collaborative activities
- Entry-level contests
- Spectators are essential, not peripheral

Real-time Feedback



- Real-time judging leads naturally to real-time feedback
- On-demand scoring
 - "Test My Solution Now"
- Leads to various flavours of responses

Real-time Feedback Responses

- Incremental scoring
 - "You have passed 10 out of 15 test cases, here is what cases did not pass and what was expected"
- Resubmission
 - "You passed 10 out of 15, and if you wish, you can have the remaining 5 cases retested for X% credit"
- Risk vs. reward/information

Real-time Feedback



- "Buy" a test case
 - "Tell me the answer, if I agree not to get credit for it"
- Unscored mock judgement
 - "Hypothetically speaking, how am I doing?"
- Scoring of sample cases
 - "Do I understand the basic cases?"

Real-time Feedback



- All of these mechanisms provide more information to the student
- Provides some positive reinforcement (or, at least, redirection) for some students
- Opens up some new dimensions to competition

Test case creation task

- Consider another side of competition
 - Test case submission (correctness of test)
 - | "If I really understand this problem, on input X, I should be getting output Y"
 - | This helps the individual contestant, without influencing their score (directly)
 - Can also use testing as a way to augment scores

Test case creation task

- Test case scoring
 - Similar to TopCoder challenge phase, but not program specific
 - Score is I/N (I =incorrect found, N =total), where N may also include "error prone judges"
- Test case wiki
 - Collaborative, but opportunity for score augmentation

Graduated difficulty of tasks

- Vast majority of contestants are "wrong"
 - where "wrong" means "incorrect" meaning "partially correct"
- Larger problem sets covering a range of difficulty
 - address the 50% rule
 - address the "0 median problem"
 - allow a *natural* gradation of scores
 - still require a continuum of scores for upper tail

Graduated difficulty of tasks

- Speed of submission as an element of score
 - TopCoder is extreme
 - ACM ICPC is moderate
 - IOI is implicit (see talks from Bogdan and Michal)
 - It is enough (in our view) to code a reasonable algorithm, and this does make the contest different for the contestants

Graduated level of difficulty

- Open-ended tasks
 - sometimes called "relative scoring"
 - evaluated "against" each other
 - | Really is just a continuum of scores
 - setting a precisely defined but intractable problem are excellent "real world" examples of relative scoring opportunities
- Tournament-based evaluation

Collaborative activities



- Wiki-based activities
 - easy-to-use template
 - highly collaborative
 - geographically diverse
 - opportunity for discussion
 - open-ended possibility
 - instant archival ability
- Informatics-based leisure activities (see Sudoku)

Entry-level contests

- Informatics is not as well established as physics, chemistry, biology and math
- Informatics has
 - little formal training of high school teachers
 - typically large programs in university
 - misinformation in terms of its definition

Entry-level contest

- IDEA: Novice programming contest with no knowledge of a programming language or environment
 - learn the process of thinking like a computer scientist
 - explicitly draw out examples of interesting, accessible computer science problems
 - Australian Informatics Competition

Spectators are essential, not peripheral

- Who are the spectators?
 - Parents
 - Peers
 - Sponsors
 - Teachers
- Suspense vs. information
- Live coverage, feedback, "play along"

Summary



- Viewing the world of contests as points in a continuous space
- Augmenting the contest may
 - increase incentive for the elite and non-elite competitors
 - increase outside interest
 - add other interesting components to round out the educational experience