Bryce Sandlund

Andrew Morgan

Menghui Wang

Team Strategy:

Early (5-10 minutes)

* Divide problems equally
* Find the easiest problem
* Finder is the coder
  + Focus on accuracy for first submission. A couple minutes of looking over the code is better than a 20 minute penalty
* Others find next easiest
  + Next easiest is "on deck" and prepares to code next

Mid

* Maintain a (priority) queue of problems ranked from easiest to code -> hardest to code
  + Use scoreboard as a guide, high acceptance rate and the number of acceptances indicate an easier problem
* Active coder codes
  + Codes **as if** we were going to wait until the last minute to submit (though we won't actually)
  + After submission, if result is not *Accepted*, coder takes a maximum of 5 minutes to find a simple bug on the computer. If going to take more than 5 minutes, or 5 minutes does not yield a fix, active coder prints his code and debugs on paper, yielding the machine to "On deck"
* "On deck" prepares his solution on paper
  + If solution is prepared, works on others until **2-minute warning** (active coder is very close to submission)
* Other person adds problems to the priority queue. If he is sure he has found the next easiest, begins to prepare ideas on paper
* "On deck" and third person should agree on a rough idea of the approach to solving a problem before writing solution on paper
* Maintain progress sheet
  + e.g. A ~~B~~ C ~~D~~ ...
* If a solution is getting *WA* and coder thinks it's correct
  + Have another person look over the code
  + If both agree it should be right, move on and come back to the problem later, time-permitting

End (90-120 minutes remaining)

* Determine which problems are feasible to still be solved in the time remaining, and stop looking at the other ones
* Start to double-up or triple-up on problems
  + Should only be working on a maximum of two problems with 90 minutes remaining
  + By the last half hour, solutions should be roughly sequential. Solve one, then look at the next.