CSE 392– Programing Challenges
Prof. Steven Skiena
Spring 2012

The 100 Problem Read
Due Thursday, May 3, 2012

Learning to read and understand problems quickly is an important skill for solving them. In this assignment, you will be given a block of 100 programming problems and asked to read each one quickly. Your goal is to get a quick impression of the difficulty and interestingness of the problem, as well as the techniques which are likely to be relevant in solving it.

For each problem you review, you are to produce a one-line review of the problem in a tab-separated format with the following fields in order:

- *Problem ID* – Give the problem id, with a '*' if the problem is likely to be one of the 10-20 'best' of the problems you review. Best is your personal utility function measuring how interesting you think it is to think about, how well written it is, and the techniques requred to solve it.
- *Problem title* – Give the title of the problem, in quotes.
- *Academic level* – How advanced a student do you need to be to solve this? Freshman-sophomore should be labeled 1, junior should be labeled 2, and senior-grad should be labeled 3. The issue is not so much how hard it is, but does the problem rest on elementary material or stuff you learn in higher levels? Many very interesting and challenging problems are labeled 1 if they rest on basic material in interesting ways.
- *Interestingness* – How interesting will this problem be to solve? Here the scale I like is "fun", "ok", "dull".
- *Writing quality* – How interesting is this problem to read? Here the scale I like is "good", "average", "bad". Give points for a clear description, good humor, and decent grammar. Take off points for irrelvant distractions, lame jokes, confusion, and hard-to-read writing.
- *Topics* – Here give a comma separated list (with *no white space for parsibility*) of a few relevant categories to file the problem under. The first should reflect which "chapter/topic" of the book/course the problem should be filed under, although you can add a little more detail with other entries to hit at what the problem really is about.

  The chapter/topic should be drawn from the following list: iteration, data-structures, strings, sorting, arithmetic, combinatorics, number-theory, backtracking, graphs, dynamic-programming, grids, geometry, computational-geometry, other.

Here are examples of my reviews for three problems:

```
10053   "Envelopes"     1       ok      average iteration,bin-packing
10054*  "The Necklace"  3       fun     good    graphs,eulerian-cycle,TSP
10055   "Hashmat the Brave Warrior"     1       dull    average arithmetic,subtraction
```

The problems can be found at:

```
http://uva.onlinejudge.org/
http://icpcres.ecs.baylor.edu/onlinejudge/index.php?option=com_onlinejudge&Itemid=8
```

Each student is assigned a distinct bank of 100 problems, as identified by the last four digits of their student ID. The first set correspond to contest set problems:

volume CXII	4841
volume CXIII	9404
volume CXIV	8298
volume CXV	2849
volume CXVI	4337
volume CXVII	2608
volume CXVIII	7468
volume CXIX	8337
volume CXX	4596
volume CXXI	6171
volume CXXII	0809
volume CXXIII	1206

The second set corresponds to Problem set problems:

volume IV	9602
volume V	5011
volume VI	9064
volume VII	8505
volume VIII	4404
volume IX	4168
volume X	3859
volume XI	6600

Rules of the Game

- The grading on this will be based on whether you made a reasonable effort. Do your best and it will suffice.
- I believe each problem should take about 5 minutes to read and review, which puts the total time at about ten hours of effort. However, you will go bonkers if you try to do it all at once. Try to do it in blocks of ten problems or so to keep yourself fresh.
- Let me know if this takes you substantially more than five minutes per problem.
- I will ask you to email your completed lists to me at skiena@cs.sunysb.edu.