HW 0 (due Monday, at noon, August 31, 2015)

NEW CS 473: Theory II, Fall 2015 Version: **1.05**

This homework contains two problems. **Read the instructions for submitting homework on the course webpage.**

You also have to do quiz 0 online (on moodle).

Collaboration Policy: For this homework, each student should work independently and write up their own solutions and submit them.

Read the course policies before starting the homework.

- Homework 0 and Quiz 0 test your familiarity with prerequisite material: big-Oh notation, elementary algorithms and data structures, recurrences, graphs, and most importantly, induction, to help you identify gaps in your background knowledge. You are responsible for filling those gaps. The course web page has pointers to several excellent online resources for prerequisite material. If you need help, please ask in headbanging, on Piazza, in office hours, or by email.
- Each student must submit individual solutions for these homework problems. For all future homeworks, groups of up to two students may submit (or present) a single group solution for each problem.
- Please carefully read the course policies on the course web site. If you have any questions, please ask in lecture, in headbanging, on Piazza, in office hours, or by email. In particular:
  - **Submission**: Please submit all solutions to all questions in a single PDF file. You can upload your solution to moodle.
    Have your name and NetID clearly printed on first page.
  - **You may use any source at your disposal**: paper, internet, electronic, human, or other, but you must write your solutions in your own words, and you must **cite explicitly**[1] every source that you use (except for official course materials). Please see the academic integrity policy for more details.
  - No late homework will be accepted for any reason. However, we may forgive quizzes or homeworks in extenuating circumstances; ask the instructor for details.
  - Answering "I don't know" to any (non-extra-credit) problem or subproblem, on any homework or exam, is worth 25% partial credit.
  - Algorithms or proofs containing phrases like and so on or repeat this process for all n, instead of an explicit loop, recursion, or induction, will receive a score of 0.
  - Unless explicitly stated otherwise, every homework problem requires a proof.

1. (50 PTS.) Some probability required.

Consider an undirected graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ with n vertices and m edges, with $m \geq n$.

(A) (10 PTS.) Let $p = n/(2m)$. For every vertex $v \in \mathsf{V}$, pick it to be in the set X with probability p. What is the expected number of vertices in X?

(B) (10 PTS.) Let G_X be the ***induced*** graph by G on X. Formally, $\mathsf{V}(\mathsf{G}_X) = X$, and an edge $uv \in \mathsf{E}$ is in $\mathsf{E}(\mathsf{G}_X)$ if and only if both u and v are in X. Provide and prove an exact bound for the expected number of edges in G_X. (Hint: Linearity of expectations.)

(C) (10 PTS.) Set $Y \leftarrow X$. Next, for every edge $xy \in \mathsf{E}(\mathsf{G}_X)$, remove, say, the vertex x from the set Y (i.e., you remove at most one vertex for every edge of G_X). Let Z be the resulting set of vertices. Prove, that induced graph G_Z has no edges.

[1] For example: "I found the solution to this exercise on `http://www.endoftheinternet.com/`. Since I understand the submission guidelines, I read this solution carefully, understood it, believe that it is correct, and I wrote it out in my own words. I was, of course, not so mind boggling stupid to just cut and paste some random text I found on the internet." (Of course, you need only the first sentence.)

(D) (10 PTS.) Provide a **lower** upper bound, as tight as possible, on the expected size of Z.

(E) (10 PTS.) Prove, using the above (and only the above), that in any graph G with n vertices and m edges, there is always an independent set of size $\Omega(n^2/m)$. (Proving this argument formally is easy but surprisingly subtle - be careful.)

2. (50 PTS.) My point is shrinking. (50 PTS.)

(A) (40 PTS.) Let $\mathsf{p} = (p_1, \ldots, p_d)$ be a point in $\mathbb{R}^d$, with all the coordinate being positive integers. Consider the following fabulous algorithm.

```
shrink(p) :
  if p_1 = p_2 = ··· = p_d then return p_1
  f ← 1
  for i = 1, ..., d do
      if p_i is odd then f ← 0
  if f = 1 then
      return 2*shrink((p_1/2, p_2/2, ..., p_d/2)).        (*)
  q ← p
  for i = 1, ..., d do
      if p_i is even then
          q_i ← q_i/2
          return shrink(q)

  α ← arg min_i q_i
  β ← arg max_i q_i
  q_β = q_β − q_α
  return shrink(q)                                        (**)
```

Here is an example of the execution of **play**$\Big((14, 2048, 1022)\Big)$.

$(14, 2048, 1022) \to^* (7, 1024, 511) \to (7, 512, 511) \to (7, 256, 511) \to (7, 128, 511) \to (7, 64, 511) \to$
$(7, 32, 511) \to (7, 16, 511) \to (7, 8, 511) \to (7, 4, 511) \to (7, 2, 511) \to (7, 1, 511) \to$
$(7, 1, 510) \to (7, 1, 255) \to (7, 1, 254) \to (7, 1, 127) \to (7, 1, 126) \to (7, 1, 63) \to (7, 1, 62) \to$
$(7, 1, 31) \to (7, 1, 30) \to (7, 1, 15) \to (7, 1, 14) \to (7, 1, 7) \to (7, 1, 6) \to (7, 1, 3) \to (6, 1, 3) \to$
$(3, 1, 3) \to (2, 1, 3) \to (1, 1, 3) \to (1, 1, 2) \to (1, 1, 1) \to$ Output: 2.

Prove (maybe using induction, but you do not have to) that **shrink** always terminates.
(Hint: Come up with an argument why in each step some non-trivial progress is being made. As a warm-up exercise, prove that the algorithm always terminates if the initial input has three numbers.)

(B) (5 PTS.) Assuming that the input p is given using the (standard) binary representation, let N be the number of bits needed to represent p. Provide a tight bound, to the value of N as a function of the values of $p_1, \ldots, p_d$.

(C) (5 PTS.) Provide a bound, as tight as possible, on the running time of **shrink** as a function of N and d. (Hint: First analyze the algorithm when (**) never happened. Then, extend your analysis to the case that (**) does happen.)