- **Groups of up to three people can submit joint solutions.** Each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the Gradescope names and email addresses of each group member. In addition, whoever submits the homework must tell Gradescope who their other group members are.
- **Submit your solutions electronically on the course Gradescope site as PDF files.** Submit a separate PDF file for each numbered problem. If you plan to typeset your solutions, please use the LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera).
- Please sign up on Gradescope with your real name and your `illinois.edu` email address. Failing to use your real name and your UIUC email would lead to problems with handling your grades. You can signup to Piazza using a pseudo-name if you want to.

Some important course policies

- **You may use any source at your disposal** – paper, electronic, or human-but you *must* cite *every* source that you use, and you *must* write everything yourself in your own words. See the academic integrity policies on the course web site for more details.
- For any question, you can write **IDK** ("I Don't Know") and get 25% of the points for the question. You would get the points if and only if IDK is the only content of your answer. Any answer containing "IDK" anywhere and **any** additional text would immediately get a zero.
- **Avoid the Three Deadly Sins!** Any homework or exam solution that breaks any of the following rules will be given an *automatic zero*, unless the solution is otherwise perfect. Yes, we really mean it. We are not trying to be scary or petty (Honest!), but we do want to break a few common bad habits that seriously impede mastery of the course material.
  - Always give complete solutions, not just examples.
  - Always declare all your variables, in English. In particular, always describe the specific problem your algorithm is supposed to solve.
  - Short **complete** answers are better than longer answers. Unnecessarily long answers (which by definition are not perfect) would get zero (i.e., 0) points. Avoid empty expressions like "in fact", "as anybody, or their uncle, can see if they think about it...", etc.
  - Always give credit to outside sources! (Yes, we are no good with counting.)

See the course web site for more information.

If you have any questions about these policies,
please do not hesitate to ask in class, in office hours, or on Piazza.

Extra problems (one fully solved) is available in the HW 1 extra problems collection on the class webpage. It is recommended that you look on these extra problems before doing the homework, since it would help you with doing the homeworks. These are also good practice problems for the midterms and final.

1 (100 PTS.) Greedy coloring

Given an undirected graph G with n vertices, the greedy coloring algorithm order the vertices of G in an arbitrary order $v_1, \ldots, v_n$. Initially all the vertices are not colored. In the ith iteration, the algorithm assigns v_i the smallest color (i.e., positive integer) k such that none of its neighbors that are already colored have color k. Let $f(v_i)$ denote the assigned color to v_i.

1.A. (30 PTS.) Prove that the above algorithm computes a valid coloring of the graph (i.e., there are is no edge uv in G such that $f(u) = f(v)$).

1.B. (30 PTS.) Prove that if a vertex v is colored by color k, then there is a simple path in the graph $u_1, u_2, \ldots, u_k = v$, such that for $i = 1, \ldots, k$, we have $f(u_i) = i$ (and $u_i u_{i+1} \in \mathsf{E}(\mathsf{G})$ for all $i = 1, \ldots, k-1$).

1.C. (40 PTS.) Prove that G

(i) has a simple path with $\lfloor \sqrt{n} \rfloor$ vertices, or

(ii) G contains an independent set of size $\lfloor \sqrt{n} \rfloor$.

A set of vertices $X \subseteq \mathsf{V}(\mathsf{G})$ is ***independent*** if no two vertices $x, y \in X$ form an edge in G.

2 (100 PTS.) Prefix it.

Let $L \subseteq \{0, 1\}^*$ be a language defined as follows:

(i) $\varepsilon \in L$.

(ii) For all $w \in L$ we have $0w1 \in L$.

(iii) For all $x, y \in L$ we have $xy \in L$.

And these are all the strings that are in L. Prove, by induction, that for any $w \in L$, and any prefix u of w, we have that $\#_0(u) \geq \#_1(u)$. Here $\#_0(u)$ is the number of 0 appearing in u ($\#_1(u)$ is defined similarly). You can use without proof that $\#_0(xy) = \#_0(x) + \#_0(y)$, for any strings x, y.

3 (100 PTS.) A recurrence.

Consider the recurrence

$$T(n) = \begin{cases} T(\lfloor n/3 \rfloor) + T(\lfloor n/4 \rfloor) + T(\lfloor n/5 \rfloor) + T(\lfloor n/6 \rfloor) + n & n \geq 6 \\ 1 & n < 6. \end{cases}$$

Prove by induction that $T(n) = O(n)$.

(An easier proof follows from using the techniques described in section 3 of these notes on recurrences.)