Extra Credit Options

No Collaboration. **Do Problem Set No. 6 First!**
Do at most one of the programming assignments (1 and 2)

1. Write a computer program, as efficient as you can make it, to test a Boolean formula for satisfiability. Test your program on various satisfiable and nonsatisfiable formulas. Write a short paper describing your algorithm, your implementation, your experimental results, and your conclusions.

2. Write computer programs, as efficient as you can make them, to implement one or more in-place algorithms to sort an array of n single-precision integers. Examples of such algorithms include quicksort, heapsort, binary insertion sort, samplesort, and various hybrid methods. Explore worst-case and average-case running times by running your programs on various data sets, measuring number of comparisons, number of data moves and overall running time. Write a short paper describing your algorithm, your implementations, your experimental results, and your conclusions.

3. Prove or disprove: the following Boolean formula is satisfiable:

\[
\begin{align*}
& (x_3 \lor \neg x_{16} \lor x_{18}) \land (x_5 \lor x_{12} \lor \neg x_9) \land (\neg x_{13} \lor x_2 \lor x_{20}) \\
& \land (x_{12} \lor \neg x_9 \lor \neg x_5) \land (x_{19} \lor \neg x_4 \lor x_6) \land (x_9 \lor x_{12} \lor \neg x_5) \\
& \land (x_1 \lor x_4 \lor \neg x_{11}) \land (x_{13} \lor x_2 \lor \neg x_{19}) \land (x_5 \lor x_{17} \lor x_9) \\
& \land (x_{15} \lor x_9 \lor \neg x_{17}) \land (\neg x_5 \lor \neg x_9 \lor \neg x_{12}) \land (x_6 \lor x_{11} \lor x_4) \\
& \land (\neg x_{15} \lor \neg x_{17} \lor x_7) \land (\neg x_6 \lor x_{19} \lor x_{13}) \land (\neg x_{12} \lor \neg x_9 \lor x_5) \\
& \land (x_{12} \lor \neg x_1 \lor x_{14}) \land (x_{20} \lor \neg x_3 \lor \neg x_5) \land (x_{10} \lor \neg x_7 \lor \neg x_8) \\
& \land (\neg x_5 \lor x_9 \lor \neg x_{12}) \land (x_{18} \lor \neg x_{20} \lor \neg x_3) \land (\neg x_{10} \lor \neg x_{18} \lor \neg x_{16}) \\
& \land (\neg x_1 \lor \neg x_{11} \lor \neg x_{14}) \land (x_8 \lor \neg x_7 \lor \neg x_{15}) \land (\neg x_8 \lor x_{16} \lor \neg x_{10})
\end{align*}
\]

4. (Modified cycles within cycles decomposition)

Prove that any strongly connected graph G is either a single vertex or can be decomposed into a set of two or more vertex-disjoint strongly connected graphs G_0, G_1, G_2, ..., G_{k-1} such that there is at least one edge e_i from G_i to G_{(i+1) \mod k} for each 0 \leq i < k, and every edge in G is either in some G_i or leads from some G_i to G_{(i+1) \mod k}.
Each such $G_i$ can be decomposed in the same way, recursively. Describe an efficient algorithm to apply this decomposition recursively to any strongly connected graph (decomposing it all the way down to single vertices) and analyze the running time of your algorithm.

5. An undirected graph $G$ with $n$ vertices is a $c$-expander if, for any set of vertices $S$ such that $|S| \leq n/2$, there are at least $c|S|$ vertices not in $S$ adjacent to at least one vertex in $S$. Prove that any $c$-expander is connected and indeed has a diameter of $O(\log n)$, where the constant depends on $c$. (The diameter is the maximum distance between any two vertices, counting each edge as length one.) What is the dependence of the $O(\log n)$ diameter bound on $c$?