CS 330 – Spring 2016, Assignment 7

Problems due by 7PM on Thursday April 28th, in the dropbox

Question 1 (10 pts). *Chapter 8, Exercise 16.*

Question 2 (10 pts). *Chapter 8, Exercise 17. [Hint: reduce from another NP-complete problem involving cycles.]*

Question 3 (10 pts). *Chapter 11, Exercise 10.*

Question 4 (20 pts). *In this programming question, we will be revisiting the* NUMBER PARTITION *problem we saw in Homework #5, this time attacking it with local search heuristics.*

Recall that a candidate solution $S = (s_1, s_2, \ldots s_n)$ *for an instance of* NUMBER PARTITION *is a sequence of signs* $s_i \in \{-1, +1\}$ *yielding a* residue $r(S) = |\sum_{i=1}^{n} s_i a_i|$*. A* random *solution is one such sequence where each of the* s_i*'s are chosen to be -1 or +1 independently at random.*

Local search operates on neighborhoods $N(S)$ *of solutions* S*: one natural definition of neighbors of* S *is all solutions that differ from* S *in one or two signs* s_i*. If we let* A_{-1} *and* A_{+1} *be the subsets of indices set to -1 and +1 respectively for a given solution* S*, then any neighbor of* S *corresponds to either moving one or two indices from one subset to the other, or swapping a pair of indices between* A_{-1} *and* A_{+1}*. A* random move *on this state space can be defined as follows: Choose two random indices* i *and* j *from* $[1, n]$ *with* $i \neq j$*. Set* s_i *to* $-s_i$ *and with probability 1/2, set* s_j *to* $-s_j$*.*

You will now try four heuristics for generating solutions to this problem:

- *Karmarkar-Karp: As defined and implemented in Homework #5.*
- *Repeated Random: Generate* k *random solutions to the problem. Return the one with smallest residue.*
- *Gradient Descent: Generate an initial random solution* S*. Then for each of* k *iterations, consider a random move (as defined above) to a neighbor* S' *of the current solution* S*. If the residue* $r(S')$ *is less than* $r(S)$*, set* S *to* S'*. Return the final (and best)* S*.*
- *Simulated Annealing: Proceed as with gradient descent (always accepting better random moves) but also accepting worse moves, i.e.* $r(S') > r(S)$*, with probability* $e^{-(r(S')-r(S))/T(i)}$*, where* $T(i)$ *is the temperature at iteration* i*. Return the smallest residue seen over all iterations (the final one may not be best). We suggest using the cooling schedule* $T(i) = 10^{10}(0.8)^{\lfloor i/300 \rfloor}$*, but you can experiment with this as you please.*

Generate fifty random instances of sets of 100 integers chosen uniformly from $[1, 10^{12}]$*. Then run each of your heuristics on these 50 test cases, using a number of iterations* k *of at least 25,000. Produce tables and/or graphs clearly demonstrating the results. Briefly compare and discuss the performance of each of the heuristics.*

You may work in teams of up to three people for this question. Teams of two are probably ideal. Each team must submit one hard copy of their code along with one joint writeup for this question.

Question 5 (Extra Credit: 10 pts). *Chapter 8, Exercise 34. [Hint: Don't be overly intimidated by the long writeup. Consider using a reduction from* HITTING SET.]