Homework 9

Huimin He , section 1

March 13, 2015

1. 9.1 HW (Knapsack decision problem NP-complete)
   (a) Input: value parameter $M > 0$, list of values $[v_1, ..., v_n]$, list of weights $[w_1, ...w_n]$, weight limit W.
   Question: does a subset $S \subseteq \{1, ..., n\}$ exist s.t. $\sum\limits_{k \in S} v_k \geq M$ and $\sum\limits_{k \in S} w_k \leq W$.

   (b) witness: a subset $S \subseteq \{1, ..., n\}$
   Relevant facts: $\sum\limits_{k \in S} v_k \geq M$ and $\sum\limits_{k \in S} w_k \leq W$.

   c) to show that KNAP is NP - complete, we need to show that $KNAP \in NP$ and construct the following reduction
   $$SUBS \preccurlyeq_p KNAP$$
   To reduce an instance of SUBS to KNAP, create the following KNAP problem let
   $$v_i = w_i = s_i$$
   where s_i is the integer numbers in the subset sum problem.
   Subset sum problem: Instance: Non-negative integer numbers $s_1, s_2, ..., s_n, andt$
   Question: Does a subset of these numbers add up to t?

   Denote b to be the weight limit, k to be the profit that we want to ask if the knapsack has value at least k.

   contruction:
   $$v_i = w_i = s_i$$
   $$t = b = k$$
   Then the knapsack problem is reduced to the subsetsum problem because
   $$\sum_{i \in S} w_i \leq b$$
   $$\sum_{i \in S} v_i \geq k$$
   $$\iff \qquad \sum_{i \in S} s_i = t$$
   Yes answer to the subset sum problems gives the left part of the relationship and satisfies the instance for knapsack problem.

2. 9.2(NP-hard problems)
   (a) B implies A so (a2) is correct. Because cook reductions needs an oracle to solve membership in L_2 in order to solve membership in L_1 but there is no oracle in Karp reduction.

   (b) Integer Knapsack problem
   Input: list of values $[v_1, ..., v_n]$, list of weights $[w_1, ...w_n]$, weight limit W.
   Optimization Objective: Find a subset $S \subseteq \{1, ..., n\}$ to maximize the $V_max = \sum\limits_{k \in S} v_k \geq M$ under the constraint that $\sum\limits_{k \in S} w_k \leq W$.

   Output: V_max

   (c)Prove that integer knapsack is np-hard by cook-reduction from KNAP.
   Denote the integer knapsack optimization problem as KNAP-opt. We only need to show that KNAP $\preccurlyeq_c ook$ KNAP-opt in order to show Knap-opt is np-hard because KNAP is np-complete from HW 9.1.

   Algorithm to solve KNAP-opt calling KNAP: We can apply binary search to find the V_max.
   Refer the KNAP algorithm to HW9.1
   First choose value M to be V_total,the sum of value list $[v_1, ..., v_n]$ run KNAP
   If the answer is yes, increase M by $(upbound - M)/2$ and run KNAP again. Also update $lowerbound = M$
   If the answer is no, decrease M by $(M - lowerbound)/2$ and update $upperbound = M$ and run KNAP again

   The oracle needs to be called at most $log_2 V_{total}$ times.

3. 9.3 Hamiltonian path To show that HAMILTON-PATH is NP-hard, we need to show that HAMILTONIAN $\preccurlyeq_{cook}$ HAMILTON-PATH.

   Pseudocode: solve HAMILTONIAN USINGHAMILTON-PATH oracle.
   Input: a graph G Output: yes/no answer

   Initilization
   add a new vertex u to G
   pick a vertex $v \in G$
   connect all vertices that are connected to v to u
   return HAMILTON-PATH on the new graphG'

   Correctness of this reduction: hamiltonian cycle can be converted to path in G' by starting at v until the cycle reaches a vetex s befroe returning to v. Finish at u instead of v.

   Hamitonian path from v to u in G' can be converted to a Hamiltonian cycle in G by starting with v and when it reached s before u, return to v instead.

4. 9.4 Metrix traveling salesman
   (a) given a graph G described in problem. Is there a hamilton cycle with cost less than k?

   witness: a hamilton cycle in G with cost less than k. This can be verified easily by checking weather it is a hamilton cycle and adding up the total cost to see if it is less than k.

(b) reference to the text book page 1097
Show that HAMTONIAN $\preccurlyeq_p$ MTS. Denote $G = (V, E)$ as an instance of HAMTONIAN. We need to construct an instance of MTS. From the complete graph $G' = (V, E')$, $E' = (i, j) : i, j \in V and i \neq j$, define the cost function c by
$c(i, j) = 0$ if $(i, j) \in E$
$c(i, j) = 1$ if $(i, j) \notin E$

Graph G has a hamiltonian cycle if and only if graph G' has a tour of cost at most 0. Suppose that graph G has a hamiltonian cycle h. Each edge in h belongs to E and has cost 0. Suppose graph G' has a tour h' of cost at most 0. Because the cost s of the edge in E' are $0, 1$. So h' has only edges in E. So h' is a hamiltonian cycle in G.

5. 9.5 (amortized analysis queue via stack)
(a)loop invariant: totle number of element in S and R does ont change. If we concatenate R stack and reverse S stack , the order of the elements does not change. The last statement makes sure that S eventually has elements from R in reverse order.

6. 9.6 (MAX 3 sat problem)
(a1) sameple space: $\{0, 1\}^n$ since we have n variables.
(a2) random variable: X is the number of satified clauses. let I_i be the value of the ith clause. So

$$X = \sum_i I_i$$

So

$$E(X) = E \sum_i I_i = \sum_i E(I_i)$$

$$E(I_i) = 1 \times P(I_i = 1) + 0 \times P(I_i = 0) = 1 \times 7/8 + 0 \times 1/8$$

Where $P(I_i = 1) = 1 - (1/2)^3 = 7/8$

(b) prove that $m \geq 7s/8$ is tight
Consider the contrsuction of the clauses as following
we have $8n$ clauses where n is positive integer. Pick first $7n$ randomly from all the variables. Then pick $1n$ clauses such that each clause is a negation of one of the first $7n$ clauses. By doing this, we make sure that the number of clauses that are correct is at most$7/8n$ because the $1n$ contradicts the other $7n$ clauses.

(d1) We can treat the process as geomteric process with $p(success) = 1/(s + 1)$ when $X \geq 7s/8$. let N denote the expected number of trials needed untill one success. $N \sim$ geometric distribution.

$$E(N) = 1/p(success) = (s + 1)$$

(d2) The randomized algorithm runs in polynomial expected time. It needs $s + 1$ expected trials to achieve $X \geq 7s/8$.