Assignment 6

Yang David Zhou, ID 260517397

December 2, 2014

Problem 1. *Hitting Set.*

Solution. Use this algorithm:

For all m $B_i = a_1, ..., a_c$, remove each one of the a_i from A and recursively call the algorithm with $k = k - 1$. If at least one of the a_i in each of the B_i returns a $k - 1$ size hitting set then there is a k size hitting set.

The running time is $T(m, k) = c \cdot (T(m, k - 1) + m)$ which is bounded by $c^k \cdot c \cdot k \cdot m \cdot$. So $f(c, k) = c^k \cdot c \cdot k$ and $p(n, m) = m$.

This algorithm works because, $\exists H, |H| = k$ if and only if $\exists a_i \in B_i \forall B_i \subseteq A$ such that removing a_i from this instance of the problem can give a new hitting set $H', |H'| = k - 1$. This is true because

Problem 2. *3-SAT.*

Solution. Given the constraint that each variable in this class of 3-SAT problems must appear once in exactly 3 clauses and every clause contains exactly 3 variables, there are an equal number of clauses are variables.

We can represent this problem as a graph G. Let $X = x_1...x_n$ be the set of variables and $K = k_1...k_m$ be the set of clauses. Create a node v_i for every $x_i \in X$ and a node c_j for every $k_j \in K$ and an edge (v_i, c_j) if x_i appears in k_j. G is bipartite as there are no edges between within the v_i or c_j. G contains a perfect matching. This is true because if G did not contain a perfect matching then there must be at least one x_i that does not appear in exactly 3 clauses which is a contradiction. Since G contains a perfect matching, then every variable can be set to whatever satisfies it's perfect matching clause. Since the matching is perfect, all clauses are satisfied. This satisfying assignment can be found in poly-time as finding a perfect matching is poly-time in input size and building the graph is poly-time in input size.

Problem 3. *Claws.*

Solution. We can represent this problem as a kind of 2-colouring problem. Start with any 2 colour assignment of the nodes. Check if the 2 colour assignment contains k claws, so the central vertex x is colour 1 and there are 3 adjacent vertices y_1, y_2, y_3 of colour 2. If there are k claws then we have found the maximum packing. If there are not, then we try a new colour and repeat.

This randomised algorithm has a high probability of success because there are at most $\binom{n}{2} = \frac{1}{2}n \cdot (n-1)$ colour assignments. Any random assignment is correct with probability $P(trials) = trials \cdot \frac{2}{n(n-1)}$. We are guaranteed to find the correct assignment after $\frac{1}{2}n^2 - n$ trials.

This algorithm will succeed after a polynomial number of runs and likely before all the possible colourings are tested.

Problem 4. *Monotone QSAT.*

Solution.

Problem 5. *Geography.*

Solution.

Problem 6. *Randomised Minimum s-t Cut.*

Solution.