CSE 347 Analysis of Algorithms March 20, 2019

Lecture 16 Summary

Instructor: Brendan Juba

1. The Traveling Salesperson problem is NP-complete: Given n cities and a *distance* $d(u, v) > 0$ to travel from u to v for each pair, is there a sequence of visiting all n cities and returning to start that traverses total distance at most D?

   (a) We actually show the following simpler problem, Hamiltonian Cycle is NP-complete: given a directed graph G, is there a cycle that visits each vertex exactly once?

      i. Hamiltonian Cycle reduces to Traveling Salesperson since we can create d that assigns distance 1 to edges of the graph and distance 2 to all other pairs of vertices.

      ii. Hamiltonian Cycle is in NP since we can check that a list of vertices only follows the edges and contains $|V|$ distinct vertices.

   (b) For a m clause 3CNF, the reduction creates a chain of $3m + 3$ vertices for each variable x_i, that can be traversed in both directions. For each jth clause, there is a vertex that can be reached from the $3j - 2$th vertex of the ith chain if x_i appears in the clause, and from $3j$ if $\bar{x}_i$ appears in the clause, and in either case returns to the $3j - 1$th vertex in the chain. Both ends of the ith chain have edges to both ends of the $i + 1$th, and there is a vertex s with edges to both ends of the first chain, and a vertex t with edges from both ends of the last chain. There is a single edge leaving t, to s.

   (c) If there is a satisfying assignment to the 3CNF, we can take the chain in ascending order if $x_i = 1$ and descending order otherwise; we can then take the "detour" edge to the jth clause's vertex from the ith chain if it is the first satisfied literal in the jth clause, and return to the chain afterwards. Observe that this traverses all of the chains and hits all of the clause vertices because the formula was satisfied, so it is a Hamiltonian Cycle.

   (d) Conversely, if there is a Hamiltonian Cycle, we argued that when the cycle takes a detour edge to some clause vertex, it must return to the $3j - 1$th vertex of the chain; otherwise, since it has to visit the other vertex with an edge to $3j - 1$ in order to reach $3j - 1$, the tour could not continue to a new vertex at that point. Thus, we can assign $x_i = 1$ if the chain is visited in ascending order and $x_i = 0$ otherwise, and since we managed to hit the clause vertices, this must be a satisfying assignment for the 3CNF.

2. The 3-Coloring problem is NP-complete: given an undirected graph G, is there an assignment of three "colors" to the vertices of G so that no edge joins two vertices that received the same color?

   (a) 3-Coloring is in NP since, given a list of colors for the vertices, we can scan the list of edges and look up the colors in this list, verifying that the colors of the endpoints are distinct in time $O(|E|)$.

   (b) We reduce from 3SAT by identifying the three colors with the two truth values and a third "base" value. We create a triangle in the graph that must take the three distinct

colors, and label the vertices of the triangle 'T', 'F', and 'B', fixing these names to the colors they receive.

(c) We create a vertex for each literal, and join the vertices for x_i and $\bar{x}_i$ by an edge so that they cannot take the same color; we connect both to 'B' so that they must take colors for truth values.

(d) We then create a graph structure for each clause that is designed to be not colorable if all of the literals in the corresponding clause take color 'F', and otherwise is always colorable.

(e) Thus, if we color the literals' vertices according to a satisfying assignment, these structures are each colorable so the graph is 3-colorable. Conversely, if the graph is 3-colorable, then the literals must take truth-value colors, and the literals for each clause cannot all take color 'F', so the assignment given by the colors of the literals is a satisfying assignment.