Thomas Hart

CS 252

HW #7

1. Make a 2-tape turing machine that takes as input the encoding of a directed graph. Tape one holds the encoding and tape two contains the path of the graph. Find the given start node and mark it on the second tape, which initially has all the unmarked nodes. Find all destination nodes using the edges from any marked nodes and mark all the destination nodes. Repeat until no new nodes are marked. If v_j is marked, there is a path from v_i to v_j, so accept. Else, reject.

2. Make a 3 tape TM M that checks if i equals 2^n for some n. The first tape takes in the input. Tape 2 has the unary representation of i. Check the edge case and accept if $i = 1$. Otherwise, mark every other 1 on tape two. If the last 1 is not marked, reject as i is an odd number. If it is even, use tape 3 to cut tape 2 in half essentially. Then cut that in half again and put it on tape 2. Repeat and check each time if there are an even number of 1's. If they are odd, reject. If the cuts ever result in a single 1 on a tape, $i = 2^n$ for some n, so accept.

3. Make a two-tape turing machine that runs a string x on M_1 and M_2 simultaneously. If M_1 accepts, L_1 is decidable, so accept. Else, reject. (M_1 and M_2 respectively accept L and L'.) We run them at the same time to avoid a looping issue on one or the other.
4. Make a TM B that runs M_1 and M_2, which accept decidable language L_1 and recognizable language L_2 respectively. Run string x on B, and if it accepts on both M_1 and M_2, accept. Else, reject.