

CSE 101 Homework 7
Spring, 2013, Due Thursday May 23
Divide-and-Conquer, Back-tracking
80 points total = 4 % %

Multiplication in threes Describe and analyze a divide-and-conquer algorithm for integer multiplication based on partitioning up each input into 3 blocks of digits. Is this better than the divide and conquer approach in class? (20 points, correct algorithm; 20 points efficiency).

Back-tracking: Hamiltonian path Consider the following algorithm for deciding whether a graph has a Hamiltonian Path from x to y , i.e., a simple path in the graph from x to y going through all the nodes in G exactly once. ($N(x)$ is the set of neighbors of x , i.e. nodes directly connected to x in G).

1. $HamPath(G, x : node, y : node)$
2. If $x = y$ is the only node in G return *True*.
3. If no node in G is connected to x , return *false*.
4. For each $z \in N(x)$ do:
 5. If $HamPath(G - \{x\}, z, y)$, return *true*.
6. Return *false*

- a. Explain (informally) why this algorithm is correct. (5 points)
- b. If every node of the graph G has degree (number of neighbors) at most 3, how long will this algorithm take at most? (15 points) (Hint: you can get a tighter bound than the most obvious one.)

Implementation: 20 pts Implement a back-tracking algorithm for maximum independent set (such as from class). Run your algorithm on random graphs with edge probability $1/2$ (as in the greedy algorithms assignment) for n as many different powers of 2 as you can (without using more than an one hour computer time on any one instance). How does the actual maximum independent set size compare to the size found by the greedy heuristic last assignment?