Examination "Breadth" – Theory

September 14, 2004 (proposal)

SOLVE EACH PROBLEM ON A SEPARATE PAGE. WRITE ONLY ON ONE SIDE OF A PAGE. WRITE YOUR CODE AT THE TOP OF EACH PAGE.

1. Give the definitions of NP, and of polynomial-time reductions. Show that nondeterministic polynomial time (NP) is closed under polynomial time reductions.

2. Give the definition and provide an example for undecidable problems (or formal languages). Sketch a proof that your example is indeed undecidable.

3. Consider a hash table T with m slots. Suppose that collisions are resolved by open addressing. That is, storage is controlled by a hash function $h(k, i)$ where k is the key and i is an integer. Initially we attempt to store the element in $T[h(k, 0)]$. If a collision occurs, then we attempt to store the element in $T[h(k, 1)]$, then $T[h(k, 2)]$, and so on.

   (a) In the following two addressing schemes $h_1(k)$ and $h_2(k)$ are arbitrary hash functions:

      i. Linear probing: $h(k, i) = (h_1(k) + i) \pmod{m}$.

      ii. Double hashing: $h(k, i) = (h_1(k) + ih_2(k)) \pmod{m}$.

      Compare and contrast these two hash functions. Discuss their effects on the efficiency of insertion and searching in the hash table. Where appropriate, give examples in which one hash function works better than another.

   (b) How can an element be efficiently deleted from a hash table that uses open hashing? Explain how searching and insertion must be modified to allow for deletions. In general it is desirable that the search time in a hash table depends on the load factor – the ratio of the number of elements in the table to the number of slots in the table. Is this the case when you do deletions from a hash table with open hashing? Why or why not?

4. Decribe and provide pseudocode for $Euclid\_gcd(a, b)$ that implements the Euclidean algorithm for finding the greatest common divisor of two integers a, b with $a > b \geq 0$.

   Prove that the running time of your algorithm measured as the total number of arithmetical operations (such as mod) and expressed as a function of b is $O(\log b)$.

5. Solve the following recurrence relation:

$$r_0 = 1, r_1 = 0; \quad r_n = 2r_{n-1} + 8r_{n-2} \text{ for } n \geq 2$$

   Explain all steps of the method you used. What are the values of r_4 and r_5?

SELECT ONE OF PROBLEMS. SOLVE ONLY ONE OF THESE. WRITE ON YOUR SELECTION. DO NOT INCLUDE SOLUTIONS OF MORE THAN ONE PROBLEM, ONLY THE PROBLEM LISTED ON PAGE ONE WILL COUNT.

(The only requested topic is *Graph Algorithms.*)

6. Let $G = (V, E)$, be an undirected connected graph represented as the adjacency list for its vertices. Use DFS to design an efficient algorithm that finds a path in G that goes through each edge exactly once in each direction. Provide a narrative description and pseudocode for your algorithm, justify its correctness and analyze the running time of your solution in terms of the size of G's representation. Show a small example.