Qualifying exam in theoretical computer science and design and analysis of algorithms

1. Is it recursively decidable for context free grammars G_i if

$$L(G_i) \quad = \quad \overleftarrow{L(G_i)}?$$

(The language on the right hand side is the "reverse".) Prove your answer.

2. Let R_i, G_i, M_i denote respectively regular sets, context-free grammars, and Turing machines. State for each of the following if it is recursive, recursively enumerable (r.e.), or not r.e. Give a short justification for your claims.

$$\begin{array}{l}
\{(G_i, R_j) \ : \ L(G_i)/R_j \quad \text{is a CFL}\} \\
\{(R_i, M_j) \ : \ R_i/L(M_j) \quad \text{is regular}\} \\
\{(R_i, M_j) \ : \ R_i/L(M_j) = \emptyset\} \\
\{(G_i, G_j) \ : \ L(G_i) \cap L(G_j) = \emptyset\} \\
\{(G_i, R_j) \ : \ L(G_i) \cap R_j \neq \emptyset\} \\
\{(G_i, R_j) \ : \ L(G_i) \subseteq R_j\} \\
\{(G_i, R_j) \ : \ R_j \subseteq L(G_i)\} \\
\{(M_i, M_j) \ : \ L(M_i) = L(M_j)\} \\
\{(M_i, M_j) \ : \ L(M_i) \neq L(M_j)\}
\end{array}$$

Qualifying exam in theoretical computer science and design and analysis of algorithms

3. An $n \times n$ "Hankel" matrix is a matrix A with the property that

$$A[i,j] = A[i-1, j+1], \quad 2 \leq i \leq n,\ 1 \leq j \leq n-1,$$

that is, a matrix which is constant along every anti-diagonal.

   (a) Given an arbitrary matrix B and a Hankel matrix A, both of dimension $n = 2^k$, find a divide-and-conquer algorithm for computing the matrix product AB which uses $O(n^{2.59})$ arithmetic operations.

   (Hints: $2.59 = 1 + \log_2 3$. Do the 2×2 case first. Consider the computation of Av, for a column vector v.)

   (b) Use the FFT to give an $O(n^2 \log n)$ algorithm for the same problem.

4. Given a direct graph $G = (V, E)$, we say $K \subset V$ is a "kernel" for G if:

   (a) No two vertices in K are joined by an arc in E.

   (b) For every $v \in V - K$ there is a $u \in K$ for which $(u, v) \in E$.

   Show that the decision problem "Does G have a kernel?" is NP-complete. (Hint: use $3SAT$ and the following gadgets:

   (a) For each $3SAT$ variable x_i a *2-cycle*:

   (b) For each clause $C_j = z_{j,1} \wedge z_{j,2} \wedge z_{j,3}$ a *triangle*:

Qualifying exam in theoretical computer science and design and analysis of algorithms

5. We would like to implement the abstract data type **priority dequeue**, which supports the following operations: *makedequeue, insert(key), deletemin, deletemax.*

   Show how this can be done so that *insert, deletemin*, and *deletemax* all run in time $O(\log n)$ and space n on a collection of n entries (we will not be considering *makedequeue*), using a "twin heap" structure in which each node i stores two of the items x_i, y_i.

   The proper invariant to maintain at each node i is $x_i \leq y_i$, and $x_i \leq$ *every x item below i*, and $y_i \geq$ *every y item below i*. The twin heap should be maintained as an implicit full binary tree in an array of length n.

6. The "rehashing strategy" entails, once a hash table becomes more than half full, the allocation of a new table twice as large and the rehashing into the new table of the entire old table. Show for any standard closed hashing algorithm that rehashing gives an amortized $O(1)$ expected insertion cost. Give a formal amortization analysis using a potential function. Assume that the expected cost of insertion into a table with load factor $< .5$ is ≤ 2.