CRN 31380  CIS 315  Spring 2008
Introduction to Algorithms  Midterm Exam

Numbers in parantheses indicate maximum points, approximately equal to the time (in minutes) to solve the corresponding problem. Write solutions and their justification to the four problems in the space provided. If necessary, use extra sheets for supporting arguments. Please print your name in the space above.

Midway through the exam, Allen pulls out a bigger brain.
1. Given a list \( A \) of \( n \) elements (with repetitions), an element that occurs more than \( \frac{n}{2} \) times is called the *majority element*. The subject of this problem is finding the majority element (if one exists) in linear time using the following loop:

\[
\text{While there are two elements of different values in } A \\
\text{do remove them from } A.
\]

(a) [3] **Warm-up:** Execute the loop on each of two lists representing the following multisets:
\( A \) = \{4, 3, 2, 4, 2, 5, 6, 2, 2, 2\} and \( B \) = \{a, b, c, d, e, e\}

\[
\text{Iteration 1: } A = \quad B = \\
\text{Iteration 2: } A = \quad B = \\
\text{Iteration 3: } A = \quad B = \\
\text{Iteration 4: } A = \quad B = \\
\text{Iteration 5: } A = \quad B = \\
\]

(b) [7] Formulate a useful loop invariant of the loop. (**Hint:** Let \( i \) be the number of remaining elements and \( x \) be the majority element of \( A[1..n] \).)

(c) [5] Prove the correctness of an algorithm that maintains the invariant by completing the “check list”

**Initialization:**

**Maintenance:**

**Termination:**

**Extra Point:** Propose an efficient implementation of the algorithm. What if there is no majority element?
2. This problem requires an efficient implementation of a data structure. In your answer, you should (in the provided space) name the pertinent data structure and its operations, describe an implementation, and finally argue for the resultant total computational complexity of your answer. The question is:
How fast can you make the Prim’s algorithm for Minimum Weight Spanning Tree run in the case of all edge weights in a graph being integers in the range from 1 to \( W \), where \( W \) is a constant independent from the size of the graph?

(a) [5] **Data Structure:**

(b) [10] **Implementation:** (of data structure’s operations)

(c) [5] **Complexity of Prim’s algorithm:**
3. Given a graph $G$ with $n$ vertices, the Floyd-Warshall algorithm computes all-pairs shortest distances. To construct a shortest path for every vertex pair, one may want to augment the code of the algorithm by information about the vertices on such a path. Define $\varphi^{(k)}_{i,j}$ to be the highest-numbered intermediate vertex on a shortest path from $i$ to $j$ with all intermediate vertices in the set $\{1, \ldots, k\}$.

(a) [8] Give a recursive formulation for $\varphi^{(k)}_{i,j}$. (Hint: Assume $\varphi^{(1)}_{i,j} = NIL$.)

(b) [6] Add a line to the body of the innermost loop of Floyd-Warshall to compute $\varphi^{(k)}_{i,j}$.

$$\text{for } k := 1 \text{ to } n \text{ do } \text{for } i := 1 \text{ to } n \text{ do } \text{for } j := 1 \text{ to } n \text{ do }$$
$$\{ d^{(k)}_{i,j} := \min(d^{(k-1)}_{i,j}, d^{(k-1)}_{i,k} + d^{(k-1)}_{k,j}); \}$$

(c) [6] Write a linear time algorithm to construct a shortest path from $i$ to $j$ given the matrix $\Phi$, where $\Phi[i,j] = \varphi^{(n)}_{i,j}$. (Hint: Think of the path represented as a binary tree rooted at $\Phi[i,j]$.)
4. The Matrix-Chain Product problem is that of finding an optimal parenthesization of a multi-
operand expression, with the cost measure being the total number of scalar multiplications
performed.

(a) [5] **Brute force:** Find the minimum cost of multiplying three matrices, $A, B, C$, of
dimensions $1 \times 100, 100 \times 1$, and $1 \times 100$, in this order.

(b) [5] **Complexity:** What is the complexity of the brute force evaluation of the minimum
cost of a matrix-chain product of $n$ matrices?

(c) [10] **Algorithm:** Using the dynamic programming algorithm, compute the minimum
cost of matrix-chain product of four matrices $A, B, C, D$ of dimensions $2 \times 40$, $40 \times 5$, $5 \times 60$,
and $60 \times 1$, respectively. Show the results of your computation in the matrix $\text{cost}[i,j]$ provided below,

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & & & \\
3 & & & \\
4 & & & \\
\end{array}
\]

and the corresponding optimal parenthesization of the product

$$A \times B \times C \times D$$
5. Extra Points

Prove that the ordinary matrix multiplication (defined by $(A \times B)_{i,j} = \sum_{1 \leq k \leq n} A_{i,k}B_{k,j}$) is commutative, that is $A \times (B \times C) = (A \times B) \times C$, for any three compatible matrices $A$, $B$, and $C$.

(a) Define the “inner product” of two integer vectors $a = \langle a_1, \ldots, a_n \rangle$ and $b=\langle b_1, \ldots, b_n \rangle$, $a \odot b = \min_{1 \leq k \leq n} \{a_k + b_k\}$. Prove that matrix multiplication defined by min and $+$ as $\oplus$ and $\otimes$ operations, respectively, is also associative.

(b) What is the complexity of computing the product defined in (b) above of two matrices with dimensions $p \times q$ and $q \times r$, respectively?