# MIE 235: Algorithms and Numerical Methods

**Instructor: H. Zhang**

## FINAL EXAM

Thursday, April 19, 2001
9:30 a.m. - 12:00 p.m.

---

**Last name:**

**First name:**

**Student number:**

There are *seven* questions for a total of 100 marks. The total time is 150 minutes. Answer directly on the paper. The only aid allowed is a non-programmable calculator. **Read each question very carefully.**

---

| | |
|---|---|
| **1. Multiple Choice Questions** | / 30 |
| **2. Short Answers** | / 10 |
| **3. Parametric Curves** | / 15 |
| **4. Numerical Integration** | / 10 |
| **5. Dynamic Programming** | / 15 |
| **6. Graph Algorithms** | / 10 |
| **7. Algorithm Design** | / 10 |
| **TOTAL** | / 100 |

## 1. Multiple Choice Questions [30 marks]

Each question is worth 2 marks. Choose the *single* best answer and *circle* the corresponding letter. If your answer is correct, you receive 2 marks, otherwise, you receive 0.

1. How many edges can a tree with $n$ vertices have at most?

   A. $n(n-1)$.
   B. $n(n-1)/2$.
   C. $n$.
   D. $n-1$.

2. Why is Quicksort *not* a good choice as an external sorting algorithm?

   A. Its worst-case running time is $\Theta(n^2)$, not $\Theta(n \log n)$.
   B. It is hard to implement.
   C. It may require many random accesses to array elements.
   D. The hard disk is too slow to implement recursions.

3. Which one of the following is *not* a drawback of using Lagrange polynomials for interpolation?

   A. The degree of the polynomial may be too high.
   B. The polynomials require too many multiplications to compute.
   C. Lagrange polynomial functions are not smooth enough.
   D. The interpolating polynomial may have too many wiggles.

4. Which of the following curve may *not* have a continuous second-order derivative?

   A. A piece-wise cubic Hermite curve.
   B. A piece-wise cubic B-spline curve.
   C. A third-degree Lagrange polynomial.
   D. A second-degree Lagrange polynomial.

5. Which of the following can be considered as an advantage of piece-wise Bezier curves over piece-wise Hermite curves?

   A. Bezier curves have $C^1$ continuity, Hermite curves do not.
   B. Bezier curves have $C^2$ continuity, Hermite curves do not.
   C. Bezier curves do not require explicit derivatives to be given.
   D. The basis functions for Bezier curves are simpler to compute.

6. Which of the following is one reason for which Quicksort is often preferred over Mergesort in practice?
   - A. Quicksort has a better asymptotic worst-case running time.
   - B. Quicksort has a better asymptotic average-case running time.
   - C. Quicksort is easier to analyze.
   - D. Quicksort only requires $O(1)$ extra storage space to implement.

7. The minimum key in a max-heap must
   - A. be the last element in the heap.
   - B. be the first element in the heap.
   - C. be a leaf in the heap tree.
   - D. be a left child of some node in the heap tree.

8. The minimum key in a non-empty binary search tree of distinct keys must
   - A. be a leaf in the tree.
   - B. be a left child of some node in the tree.
   - C. be a right child of some node in the tree.
   - D. not be a right child of any node in the tree..

9. How many leaves does a max-heap of $n$ nodes have?
   - A. $n - \lfloor \log_2 n \rfloor + 1$.
   - B. $n - 2^{\lfloor \log_2 n \rfloor} + 1$.
   - C. $\lfloor n/2 \rfloor$.
   - D. none of the above.

10. What is the height, asymptotically, of a Hoffman encoding tree containing $n$ letters to be encoded?
    - A. $\Theta(n)$.
    - B. $\Theta(n \log n)$.
    - C. $\Theta(\log n)$.
    - D. $\Theta(1)$.

11. What are the worst-case running times of Insert and Extract-Max in a max-heap of $n$ nodes, respectively?

A. $\Theta(n)$; $\Theta(\log n)$.
B. $\Theta(\log n)$; $\Theta(1)$.
C. $\Theta(\log n)$; $\Theta(\log n)$.
D. $\Theta(1)$; $\Theta(\log n)$.

12. Which data structure is the most appropriate to use in an implementation of breath-first search?

A. A stack.
B. A queue.
C. A binary tree.
D. A hash table.

13. When solving a system of linear equations using Gaussian eliminations and back substitutions, which one of the following will most likely occur?

A. Truncation errors.
B. Round-off errors.
C. Overflow.
D. Underflow.

14. Which one of the following is *not* a disadvantage of the Newton's method?

A. It has relatively slow convergence compared to other methods.
B. It may not converge, even though there is a root.
C. It cannot handle zero first-order derivatives.
D. It requires the computation of first-order derivatives.

15. Which of the following is an advantage of using polynomials for interpolation and approximation?

A. Infinitely integrable.
B. Compact representations.
C. Easy to evaluate.
D. All of the above.

## 2. Short Answers [10 marks]

(a) [2 marks] Express the polynomial $4x^5 + 6x^3 - 7x^2 - 2x + 9$ using the Horner's Rule.

(b) [3 marks] Given an initial array of integers 5, 12, 18, 9, 3, 19, 7, 10, in that order. Run the Quicksort algorithm with 10 as the pivot. In the space below, show what the array looks like *right before* the *first* recursion is to be performed. You do not need to show any steps.

(c) [3 marks] Show that the two curves $A(t) = (t, t^2), t \in [0, 1]$ and $B(t) = (2t + 1, t^3 + 4t + 1), t \in [0, 1]$ are $G^1$ continuous.

(d) [2 marks] Give the two conditions under which the Bisection method can be applied successfully to approximate a root of $f(x)$ over the interval $[a, b]$.

## 3. Parametric Curves [15 marks]

(a) [5 marks] Determine a parametric cubic curve $X(t)$ over [0,1] defined by the following four constraints: $X(0) = P_1$, $X(1/3) = P_2$, $X(2/3) = P_3$, and $X(1) = P_4$. You only need to give the *inverse* of the change-of-basis matrix, and show your work.

(b) [10 marks] Given the change-of-basis matrix for cubic B-splines,

$$M_{B-spline} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \qquad (1)$$

and consider two B-spline curve segments $B_1$ and $B_2$, where $B_1$ is defined by the control points $P_1, P_2, P_3$, and $P_4$, and $B_2$ is defined by the control points $P_2, P_3, P_4$, and $P_5$. Each curve segment is parameterized over [0, 1] as usual.

1. In terms of $B_1(1), B_1'(1), B_1''(1), B_2(0), B_2'(0)$, and $B_2''(0)$, where $f'$ and $f''$ denote the first and second order derivatives, give sufficient conditions for the B-spline curve consisting of $B_1$ and $B_2$ to be $C^2$ continuous.

2. Prove that the B-spline curve consisting of $B_1$ and $B_2$ has a continuous second order derivative. *Hint:* Recall that the B-spline representation of a curve segment determined by the control points $P_1, P_2, P_3$, and $P_4$ is

$$B_1(t) = T \cdot M_{B-spline} \cdot V_1,$$

where $T = [t^3 \; t^2 \; t \; 1]$ and $V_1 = [P_1 \; P_2 \; P_3 \; P_4]^t$ ($^t$ is the transpose).

## 4. Numerical Integration [10 marks]

If $f \in C^4[a,b]$, there exists a $\mu \in (a,b)$ for which the **Composite Simpson's rule for n=2m subintervals** of $[a,b]$ can be expressed with error term as

$$\int_a^b f(x)dx = \frac{h}{3}[f(a)+2\sum_{j=1}^{m-1} f(x_{2j})+4\sum_{j=1}^{m} f(x_{2j-1})+f(b)] - \frac{(b-a)h^4}{180} f^{(4)}(\mu),$$

where $a = x_0 < x_1 < \ldots < x_n = b$, $h = (b-a)/n$, and $x_j = x_0 + jh$ for each $j = 0, 1, \ldots, n$.

Use this rule to approximate $\int_3^6 \ln x \, dx$ to within $10^{-4}$. Clearly state how many subintervals are necessary. Round to four digits after the decimal in your calculations. *Hint:* The fourth-order derivative of $\ln x$ is $-6/x^4$.

## 5. Dynamic Programming [15 marks]

Consider the following **Longest Increasing Consecutive Subsequence** (LICS) problem: given a list of floating-point numbers $[v_0, v_1, \ldots, v_n]$, we want to find a consecutive subsequence of numbers $[v_i, v_{i+1}, \ldots, v_j]$ such that $v_i < v_{i+1} < \ldots < v_j$ and the number of elements in the subsequence $(j-i+1)$ is as large as possible.

For example, in the list $[3.5, -6, 4.112, 1.6, 2.7, 7, -5.01]$, $[-6, 4.112]$ is an increasing consecutive subsequence with length 2, and $[1.6, 2.7, 7]$ is an increasing consecutive subsequence with length 3. In fact, $[1.6, 2.7, 7]$ is the LICS for this example.

To solve the LICS problem using dynamic programming, we define an array $A$ such that $A[j]$ = the *length of the LICS of* $[v_0, \ldots, v_j]$ whose last element is exactly $v_j$, that is, the subsequence must have the form $v_i, \ldots, v_j$ for some $i \leq j$.

(a) [8 marks] Given the list $[3.12, 4, -2.07, 0.15, 4.25, 297, 14.5, -7]$, that is, $v_0 = 3.12$, $v_1 = 4$, $\ldots$, $v_7 = -7$, complete the following table that stores the values of $A[j]$. You need not show your work.

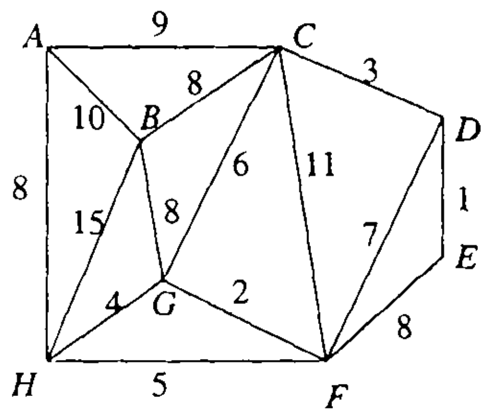
| $j$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $A[j]$ | | | | | | | | |

(b) [5 marks] Give recursive expressions for $A[0]$, and $A[j]$ for $j > 0$. You need not justify your answer.

(c) [2 marks] Suppose that you have computed the array $A$. How do you obtain the length of the LICS for the original list of numbers?

## 6. Graph Algorithms [10 marks]

(a) [5 marks] Given the weighted graph shown, use Prim's algorithm to find a minimum-weight spanning tree. Start with the vertex $A$, and list the *vertices* added to the minimum-weight spanning tree *in order*. What is the total weight of the tree you obtain?




(b) [5 marks] Besides Prim's algorithm, there is another well-know algorithm for finding the minimum-weight spanning tree of a graph, the Kruskal's algorithm. Kruskal's algorithm repeatedly adds an edge into the, initially empty, minimum-weight spanning tree it constructs, if the edge added has the *least* weight among all possible edges not yet added to the tree, and its addition into the current tree does not create a cycle.

Use Kruskal's algorithm to find a minimum-weight spanning tree of the graph shown above. Please list all the *edges* added *in order*, and when writing an edge in the form $(u, v)$, make sure that $u$ precedes $v$ in the alphabet. What is the total weight of the tree you obtain?

## 7. Algorithm Design [10 marks]

Given a list of $n$ integers, it is possible to sort them with the aid of a max-heap. Describe such an algorithm in plain English, state its worst-case complexity, and justify your claim. Of course, you should make your algorithm as efficient as possible.