

Final Exam

CSCI 539 Data Structures and Algorithms

December 17–19, 2001

Things to know before start:

(1) This exam is close-person, i.e., you should not discuss the exam with your fellow students. Please sign the honor code statement below and return your completed exam to me by 12:00pm, Wednesday, Dec. 19.

(2) The exam, however, is open-book and open-notes. You are free to consult texts, papers, and notes. Please list your sources.

(3) Read each problem carefully and make sure you understand the problems completely. I will be available to answer clarification questions.

(4) Do the easy problems first. Be sure you don't lose points on the easy ones. For hard problems that you don't have complete solutions, write down things you know to get partial credits. (Of course they have to be relevant to the problems.)

Honor Code Statement

I have neither given nor received any unpermitted aid on this examination.

(Please sign your name above)

(Please print your name above)

1. (20 points) Consider the linear-time algorithm for heap initialization.

```
Initialization(H) //H[1..n] is an array
  for i = n / 2 downto 1
    RebuildHeap(H, i)
```

- (a) Show by pseudocode how this procedure can be implemented as a recursive procedure `Initialization(H, i)`, where $H[i]$ is the root of the subheap to be built. To build the entire heap, we would call `Initialization(H, 1)`. In your recursive algorithm, you may still use `RebuildHeap`.
- (b) Give a recurrence for the worst-case time complexity of your algorithm.
- (c) Solve the recurrence in big-O.

2. (10 points) True or false? Explain briefly.

- (a) Assume that a hash table of size b stores n keys. Assume that the hash table is implemented by separate chaining with a hash function designed so that every key has an equal chance (or probability) to be stored in any of the b buckets (or linked lists). Then the worst-case time complexity to search a key in this hash table is $O(n/b)$.
- (b) A binary search tree of n integers in the range from 1 to n^2 can be constructed in $O(n)$ worst-case time.

Hint:

For (b), the binary search tree does not have to be balanced. Any binary search tree of n integers in $[1, n^2]$ will satisfy the requirement.

3. (20 points) Answer the following questions about sorting and selection:
- (a) A sorting method is *stable* if equal elements remain in the same relative order in the sorted sequence as they were in the original input. Which of the following sorting algorithms are stable?
- i. Insertion Sort (see MAW p. 254 for algorithm):
Stable _____ Unstable _____
 - ii. Shell Sort (see lecture notes for algorithm):
Stable _____ Unstable _____
 - iii. Merge Sort (see MAW p. 267 for the linear merge algorithm):
Stable _____ Unstable _____
 - iv. Heap Sort (see lecture notes for algorithm):
Stable _____ Unstable _____
 - v. Quick Sort (see lecture notes for algorithm):
Stable _____ Unstable _____
- (b) Consider the Quick Select algorithm to find the k th smallest number among n numbers. Assume that the pivot is the first element in a list. We say that an input is a *best-case* input if the Quick Select algorithm applied to the input results in the most even partition (with $|L_1| = \frac{n-1}{2}$, $|L_2| = 1$, and $|L_3| = \frac{n-1}{2}$) in every recursive call.
- i. Give the best-case input list for $n = 15$ and $k = 1$, using each number in the set $\{1, 2, \dots, 15\}$ exactly once to make up the list.
 - ii. Give the best-case input list for $n = 2^l - 1$ for some positive integer l and $k = 1$, using each number in the set $\{1, 2, \dots, 2^l - 1\}$ exactly once to make up the list.

4. (30 points) You are given the following matrix chain:

$$\begin{array}{ccccccccc}
 A_1 & \times & A_2 & \times & A_3 & \times & A_4 & \times & A_5. \\
 5 \times 10 & & 10 \times 15 & & 15 \times 8 & & 8 \times 4 & & 4 \times 10 \\
 p_0 \times p_1 & & p_1 \times p_2 & & p_2 \times p_3 & & p_3 \times p_4 & & p_4 \times p_5
 \end{array}$$

Build the table $m[i, j]$ to determine the minimum number of operations required to compute the product of the matrix chain. Also, for each $m[i, j]$ put the corresponding optimal k value next to it.

m	1	2	3	4	5
1	0				
2	<i>undefined</i>	0			
3	<i>undefined</i>	<i>undefined</i>	0		
4	<i>undefined</i>	<i>undefined</i>	<i>undefined</i>	0	
5	<i>undefined</i>	<i>undefined</i>	<i>undefined</i>	<i>undefined</i>	0

Now use the table to answer the following questions:

- What is the minimum number of operations required to compute $A_1 \times A_2 \times A_3 \times A_4 \times A_5$? _____
- Give the optimal order of computing the matrix chain by fully parenthesizing the matrix chain below.
 $A_1 \times A_2 \times A_3 \times A_4 \times A_5$
- What is the minimum number of operations required to compute $(A_1 \times A_2) \times (A_3 \times A_4 \times A_5)$? _____
- What is the minimum number of operations required to compute $(A_1 \times A_2) \times A_3 \times (A_4 \times A_5)$? _____
- What is the minimum number of operations required to compute all of the following: $A_1 \times A_2$, $A_4 \times A_5$, and $A_1 \times A_2 \times A_3 \times A_4 \times A_5$?

Hint:

For question (e), when computing $A_1 \times A_2 \times A_3 \times A_4 \times A_5$, will the results of $A_1 \times A_2$ and $A_4 \times A_5$ be useful to make the computation more efficient than computing the chain from scratch?

5. (40 points) Consider a set S of $n \geq 2$ distinct numbers given in unsorted order. Each of the following four parts asks you to give an algorithm to determine two members x and y in S that satisfy a specified condition. In as few words as possible, describe your algorithm and justify its time complexity.
- (a) In $O(n)$ worst time, determine $x, y \in S$ such that $|x - y| \geq |w - z|$ for all $w, z \in S$.
 - (b) In $O(n \log n)$ worst time, determine nonrecursively $x, y \in S$ such that $|x - y| \leq |w - z|$ for all $w, z \in S$.
 - (c) In $O(n)$ average case, determine $x, y \in S$ such that $x + y = Z$, where Z is given, or determine that no two such numbers exist. Assume that the n numbers in S are distributed uniformly in a known range.
 - (d) In $O(n)$ worst case, determine $x, y \in S$ such that $|x - y| \leq \frac{1}{n-1} (\max_{z \in S} z - \min_{z \in S} z)$.

Clarifications and hints:

- $|x - y|$ is the absolute value of $x - y$.
- $\max_{z \in S} z$ is the maximum in S and $\min_{z \in S} z$ is the minimum in S .
- The key is to use as few words as possible to describe the algorithms. You may also use the pseudocode if it gives a simpler description. Points may be taken off for wordy answers.
- You may cite any algorithms discussed in class without giving detailed descriptions.
- For (c), use hashing with separately chaining.
- For (d), use divide and conquer. Note that $\frac{1}{n-1} (\max_{z \in S} z - \min_{z \in S} z)$ is the average distance between consecutive numbers in the sorted order of S .