CS 3343 (Spring 2015) Exam 2

April 16, 2015
10:00am - 11:25am (85 minutes)

Name: ___________________________ ID: ___________________________

• Don’t forget to put your name and ID on the cover page
• This exam is closed-book
• If you have a question, **stay seated** and raise your hand.
• Please try to write legibly – if I cannot read it, you may not get credit.
• Do not waste time – if you cannot solve a question immediately, skip it and return to it later.
• Try your best to answer each question. Partial credits will be given if you show that you have some ideas – but not according to the length of your answer.
• Be succinct.

<table>
<thead>
<tr>
<th></th>
<th>Sorting and Select I</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Sorting and Select II</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Heaps</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>Longest Common Subsequence</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Fractional Knapsacsk Problem</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>Gas Station Location Problem</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Graph basics</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>(Extra credit) Algorithm Design</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td>120</td>
</tr>
</tbody>
</table>
1. (15 points) Sorting and Select Part I

Suppose that you have an array of \( n \) integers in the range \([1, k]\), and you would like to **find the median of the array**. You are mainly considering two choices: (1) using an order-statistics algorithm, such as the randomized select algorithm or the worst-case linear-time select algorithm, or (2) using a sorting algorithm to sort the array and then choose the median from the sorted array.

a. Fill in the table below with the time and memory complexities (in \( \Theta \)) of the algorithms that you can choose from. To determine the (extra) memory complexity, exclude the space for the input array.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Expected time complexity</th>
<th>Worst-case time complexity</th>
<th>Extra memory complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomized Quicksort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merge Sort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counting Sort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Randomized Select</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear-time Select</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b. When \( k \in \Theta(n) \), which algorithm(s) would you prefer and why?

c. When \( k \) is much larger than \( n \), which algorithm(s) would you prefer and why?

2. (10 points) Sorting and Select Part II. Indicate whether each of the following statement is true or false. Briefly justify your answers.

a. The best-case running time for Quick Sort is \( \Theta(n) \).

b. The worst-case running time for Randomized Quick Sort is \( \Theta(n^2) \).

c. To find the median from an unsorted array takes the same amount of time asymptotically as finding all values smaller than or equal to the median.
d. Let’s say that given an originally unsorted array, you have a program that needs to retrieve all elements, one at a time, in a non-increasing order. If no dynamic update (deletion, insertion, changing value) to the array is expected, using a heap would not provide much advantage comparing to simply sorting the whole array.

e. In theory, finding the median from an array is asymptotically much less efficient than finding the minimum value in the array.

3. (25 points) Max Heap

a. Does the array [10 8 7 4 6 3 5 1 2 0] represent a max heap? Why or why not?

b. Illustrate how BuildHeap works on an array [3 8 7 9 20 15 4]. Show necessary intermediate steps for full credit (e.g., show the content of the tree after each Heapify).

c. Professor XYZ designed a new procedure BuildHeap’ as shown below:

```plaintext
BuildHeap'(A)
    heapsize(A) = 1;
    for (i = 2 to length(A)) {
        heapsize(A) = heapsize(A)+1;
        Heapify(A);
    }
```

Will the algorithm still be able to build a max heap? Why or why not?
d. Given an array representing a **max** heap, can we reverse the order of the elements in the array to obtain a **min** heap? Why or why not?

e. Starting from the heaps on the left, show the contents of the new heaps after each heap operation.

```
15
12 9
10 9 3
6 9 1
```

- **Insert 14**

```
15
12 9
10 9 3 4
6 9 1
```

- **ExtractMax**

```
15
12 9
10 9 3 4
6 9 1
```

- **Decrease the 2nd key (12) to 5**

```
15
12 9
10 9 3 4
6 9 1
```
4. (18 points) Longest common subsequence (LCS).

a. Complete the following dynamic programming table to compute the length of LCS between two strings XYCCEXED and YXCXDXXED.

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>C</th>
<th>C</th>
<th>E</th>
<th>X</th>
<th>E</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b. What is the actual longest common subsequence? If there are multiple longest common subsequences, report all of them.

5. (7 points) Fractional knapsack problem

Given the six items in the table below and a knapsack with a weight limit of 100 LBs, what is the maximum value that can be carried with the knapsack? In order to obtain the maximum value, which items (or what fraction of them if some items need to be broken into pieces) should you put into the knapsack?

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight (LB)</th>
<th>Value ($)</th>
<th>Value/Weight ($/LB)</th>
<th>Weight(LB) taken</th>
<th>Value($) taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>50</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>40</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>20</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>10</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
6. (20 points) Gas Station Location Problem

Suppose that you are consulting for a company which plans to build gas stations along a road. Along the road there are major intersections every 2 miles, which are potentially good locations for new gas stations. The gas stations at different intersections have different estimated profits. The company has decided not to build gas stations at adjacent major intersections, i.e., two gas stations will have to be at least 4 miles away from each other. Given the estimated annual profit for having a new gas station at each intersection, you are asked to provide an optimal plan that will result in the highest estimated total profit.

a. Let \( p(i) \) be the estimated profit at intersection \( i \), and \( F(i) \) the estimated total profit of the optimal plan that considers only intersections 1 to \( i \). Write down the recurrence function for computing \( F(i) \).

b. Show how to use the recurrence function and dynamic programming to find the optimal plan given the estimated profits below.

<table>
<thead>
<tr>
<th>Intersection</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit (× $10k)</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Optimal value =
Optimal selection:

c. What is the asymptotic running time of your algorithm and why?
d. Now suppose that later you found that due to a small problem in the estimation process, the estimated revenue may not be very accurate. Specifically, you had underestimated the revenue of each location by a certain ratio (say, 10%). Do you need to rerun your algorithm to get the optimal solution? Why or why not?

e. Similar to (c), but you found that you had underestimated the revenue at each location by a fixed amount (say, $10k), do you need to rerun your algorithm to get the optimal solution? Why or why not?

7. (5 points) Graph basics.
Draw a graph that can be represented by the adjacency matrix below.

```
0 1 0 0 0 0 0
1 0 0 1 0 1 0
0 0 0 1 1 0 0
0 1 1 0 0 0 1
0 0 1 0 0 0 1
0 1 0 0 0 0 1
0 0 0 1 1 1 0
```
8. (20 points) Extra credit: Algorithm design.

a. (10 points) Nuts and bolts. You are given a collection of $n$ bolts of different widths and $n$ corresponding nuts. You are allowed to try a nut and bolt together, from which you can determine whether the nut is larger than the bolt, smaller than the bolt, or matches the bolt exactly. However, there is no way to compare two nuts together or two bolts together. The problem is to match each bolt to its nut. Design an efficient algorithm for this problem with average-case efficiency in $\theta(n \log n)$.

b. (10 points) A palindrome is a nonempty string over some alphabet that reads the same forward and backward. Design an efficient algorithm that takes a sequence $x[1 \ldots n]$ and returns the \textbf{longest palindromic subsequence}. What is the running time of your algorithm?

(For instance, the sequence ACGTGTCAAAATCG has many palindromic subsequences, including ACGCA and CAAC. On the other hand, the subsequence ACG is not palindromic.)
Blank page
Blank page