CompSci 161  Dynamic Programming Diag Exam 2, Winter 2023
DO NOT OPEN EXAM UNTIL INSTRUCTED TO DO SO
SILENCE AND STORE ALL ELECTRONICS

This is a diagnostic exam intended to help you evaluate your readiness for the real exam.

The following rules apply to you, whether you think they do or not. Read and understand them; failure to abide by these rules, or directions given by course staff during the exam, may result in disciplinary action, including but not limited to a failing grade in the class.

- This exam is solely for students enrolled in this lecture. Anyone not enrolled in this lecture may not take an exam.

- You may not open the exam or begin writing until the instructor has explicitly given you permission to do so.

- Keep your UCI ID readily accessible during the test. Proctors may request to see it.

- This exam is closed book, closed notes, and is individual effort. Once course staff begin passing out exams, you may not communicate with anyone other than proctors for any reason, nor may you have electronics, including calculators watches and phones, available to you during the test for any reason. YOU DO NOT NEED A CALCULATOR!

- If you leave your seat during the test for any reason, your instructor may collect it and deem you to have turned it in. Do not ask proctors for an exemption to this, they are not authorized to grant such.

- You must take the exam in your assigned seat unless the professor (not a TA) tells you otherwise. The instructor will call to cease writing at 11:48 AM, at which point you must immediately cease writing and close the exam. You may not write any further at that point, including finishing one’s current sentence.

- If you believe a question is ambiguous, write at least two reasonable interpretations and indicate clearly which one you will be using. Then answer your question with that assumption. Unless your interpretation makes the problem much more trivial than intended, we will grade your response as if one of us had made that clarification.

- The purpose of the real exam is to evaluate how well you understand the material presented in the course. It is an academic integrity violation to do anything that subverts the goals of this assessment including, but not limited to, not doing your own work or submitting that of anyone else.

- We will only grade responses marked in the space provided for each question.
Nothing you write on this page will be graded. The next page in this booklet contains a spot to answer these questions. You may use this page as scratch paper if you would like, and room to do so exists.

1. (2 points). Suppose I have solved a problem by dynamic programming and my recursive solution is as follows, where $A$ and $\delta$ are inputs to the problem. The top-level call is to $\text{Foo}(n, m)$.

   $$\text{Foo}(i, j)$$
   
   if $i = 0$ then
   return $j$
   else if $j = 0$ then
   return $i$
   else
   return $\min(A[i, j] + \text{Foo}(i - 1, j - 1), \delta + \text{Foo}(i - 1, j), \delta + \text{Foo}(i, j - 1))$
   end if

What will the running time of this solution be if I use a memoization vector? In which order will I fill in the vector?

2. (1 point) Consider the following table, which is the output from running the optimal binary search tree algorithm for some seven-key input. In each entry, the value printed in the upper half of the cell is $\text{Tree}[i, j]$, the cost of the optimal binary search tree consisting of keys $i \ldots j$. The value printed in the lower half of the cell is the value of $\text{roots}[i, j]$.

Suppose we have computed the following using the algorithm for the optimal binary search tree. However, we accidentally forgot to record $\text{Tree}[1, n]$. What is the missing value for that spot? It is the only one omitted in the following table.

<table>
<thead>
<tr>
<th></th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_3$</th>
<th>$k_4$</th>
<th>$k_5$</th>
<th>$k_6$</th>
<th>$k_7$</th>
<th>$k_8$</th>
<th>$k_9$</th>
<th>$k_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>0.12</td>
<td>0.3</td>
<td>0.73</td>
<td>1.12</td>
<td>1.51</td>
<td>1.8</td>
<td>2.17</td>
<td>2.45</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.09</td>
<td>0.29</td>
<td>0.41</td>
<td>0.79</td>
<td>1.09</td>
<td>1.36</td>
<td>1.73</td>
<td>2.01</td>
<td>2.25</td>
<td></td>
</tr>
<tr>
<td>$k_3$</td>
<td>0.11</td>
<td>0.23</td>
<td>0.56</td>
<td>0.82</td>
<td>1.09</td>
<td>1.46</td>
<td>1.72</td>
<td>1.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_4$</td>
<td>0.06</td>
<td>0.28</td>
<td>0.54</td>
<td>0.81</td>
<td>1.12</td>
<td>1.33</td>
<td>1.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_5$</td>
<td>0.16</td>
<td>0.42</td>
<td>0.63</td>
<td>0.94</td>
<td>1.15</td>
<td>1.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td>0.13</td>
<td>0.31</td>
<td>0.57</td>
<td>0.78</td>
<td>0.96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td>0.09</td>
<td>0.29</td>
<td>0.43</td>
<td>0.61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_8$</td>
<td>0.11</td>
<td>0.25</td>
<td>0.41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_9$</td>
<td>0.07</td>
<td>0.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{10}$</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. (1 point) Draw the optimal binary search tree for the input probabilities in the previous question.
Write your answer to question 1 between this sentence and the dotted line:

Write your answer to question 2 in the box below:

Draw your answer to question 3 between here and the end of the page.
4. (3 points) A number of languages are written without spaces between the words. Consequently, software that works with text written in these languages must address the word segmentation problem – inferring likely boundaries between consecutive words in the text. If English were written without spaces, the analogous problem would consist of taking a string like “meetateight” and deciding that the best segmentation is “meet at eight” and not “me et at eight” or “meet ate ight” or any of a huge number of even less plausible alternatives.

One approach to processing this is to find a segmentation that maximizes the total “quality” of the individual constituent words. Suppose you are given a function \( \text{quality}(S) \) that takes, as input, a string and returns a number, which may be positive or negative, where larger numbers correspond to more plausible English words. The total quality of a segmentation is determined by adding up the qualities of each of its blocks.

Use dynamic programming to design an efficient algorithm that takes as input a string \( s \) and determines the segmentation of maximum total quality. For simplicity in analyzing the running time, you may assume that the substring operation takes \( \mathcal{O}(1) \), that the quality function runs in \( \mathcal{O}(1) \), and that the quality function can (if you prefer) be \( \text{quality}(S, i, j) \) which returns the quality of \( S[i \ldots j] \), also in \( \mathcal{O}(1) \).

To complete these instructions, do the following.

- Give a clear and precise English definition that describes the function you are implementing. Not how it works (yet), but rather what it solves.
- Give that function a meaningful variable name. This is not [just] me being pedantic; I have found it helps students with this topic if they do this. “OPT” is not a meaningful variable name, nor is “table.” Single letters are not meaningful variable names.
- Give a clear recursive formula or algorithm in terms of smaller instances of exactly the same problem.
- Describe the iterative running time correctly. This can either be by writing the iterative algorithm (in which case, you can point out where the previous part is within the solution), or by taking your recursive solution, counting the cases, describing the order in which the table would be filled in, and analyzing the time accordingly.
Write your answer to question 4 on this page. It is possible that you will not use the entire page.