CS320 – Test 1 Study Review

Applied strategy

Python

Application

Logical

Complexity

Master theorem

practice

G S
Segmented Least Squares: Scientific statistical data is often plotted on a 2-Dimensional set of axes with a line that is a “best fit” through the data. The line is straight and is defined by the standard equation $y = mx + b$. The error of the line with respect to the data points is the sum of its squared distance from the points. So segmenting the points is a good idea as long as we don’t carry it too far and create a new segment for every 2 points.

There is a penalty associated with each partition which is the sum of the number of segments into which we partition $P$ times a fixed multiplier $C > 0$, and the error value of the optimal line through each segment. The goal is to find a partition of minimum penalty.
We need to think of some property of an optimal solution that will help us discover the structure of subproblems. We are trying to partition $n$ points into segments. Here is an observation: the last point $p_n$ belongs to a single segment in an optimal partition, and that segment begins at some earlier point $p_i$.

**QUESTION:** A. Fill in the blanks in the following description of potential subproblems. ONLY use the words in the following word bank.

Word bank: [add, remove, to, from, remaining, points, recursively, different]

If we knew which points are involved in the last segment, then we could _________ them _________ consideration and _________ solve the same problem on the _________ points $p_1, \ldots p_{i-1}$.
We need to think of some property of an optimal solution that will help us discover the structure of subproblems. We are trying to partition $n$ points into segments. Here is an observation: the last point $p_n$ belongs to a single segment in an optimal partition, and that segment begins at some earlier point $p_i$.

**QUESTION:** A. Fill in the blanks in the following description of potential subproblems. ONLY use the words in the following word bank.

Word bank: [add, remove, to, from, remaining, points, recursively, different]

If we knew which points are involved in the last segment, then we could ___**remove**___ them ___**from**___ consideration and ___**recursively**___ solve the same problem on the ___**remaining**___ points $p_1, \ldots, p_{i-1}$. 
QUESTION: B. How many subproblems are there?
QUESTION: B. How many subproblems are there?

1- the remaining points
QUESTION: C. Suppose we let OPT(i) mean the optimum solution for the points \( p_i \ldots p_j \). Then OPT(n) is the penalty of the last segment from \( p_i \ldots p_n \) plus the OPT of the previous segment, recursively. Clearly we will want to minimize this penalty. For the subproblem on the points \( p_1 \ldots p_j \), OPT(j) = \( \min(e_{ij} + C + OPT(i-1)) \) for all \( i \) where \( 1 \leq i \leq j \), and the segment \( p_i \ldots p_j \) is only used if this minimum uses index i. Thus, we have to look at all values of j from 1 to n, and for each j we have to compute OPT(j) for all values of i from 1 to j to find the minimum used in this recurrence.

What is the worst-case work for this part of the overall problem?

a. \( n^3 \)
b. \( \sqrt{n} \)
c. \( n^2 \)
d. \( n \log n \)
e. other
QUESTION: C. Suppose we let OPT(i) mean the optimum solution for the points \( p_i \ldots p_j \). Then OPT(n) is the penalty of the last segment from \( p_i \ldots p_n \) plus the OPT of the previous segment, recursively. Clearly we will want to minimize this penalty. For the subproblem on the points \( p_1 \ldots p_j \), OPT(j) = \( \min(e_{ij} + C + \text{OPT}(i-1)) \) for all \( i \) where \( 1 \leq i \leq j \), and the segment \( p_i \ldots p_j \) is only used if this minimum uses index \( i \). Thus, we have to look at all values of \( j \) from 1 to \( n \), and for each \( j \) we have to compute OPT(j) for all values of \( i \) from 1 to \( j \) to find the minimum used in this recurrence.

What is the worst-case work for this part of the overall problem?

a. \( n^3 \)

b. \( \sqrt{n} \)

c. \( n^2 \)

d. \( n \log n \)

e. other
QUESTION: D. What strategy/technique should you use to think about the solution to this problem?

a. divide and conquer
b. recursion
c. greedy
d. dynamic programming
e. sorting
f. other
QUESTION: D. What strategy/technique should you use to think about the solution to this problem?

a. divide and conquer
b. recursion
c. greedy
d. dynamic programming
e. sorting
f. other
QUESTION: E. How do the steps in the questions (a, b, and c) above relate to the steps of this strategy/technique? Note that the question may not completely fulfill the requirements of the approach step, but it does provide some of the work. Not all steps may be addressed by any of these questions.

Dynamic Programming:
define subproblems/characterize structure and show it is optimal _________
recursively define the optimum _________
compute the optimum storing solutions to subproblems (usually bottom-up) _________
construct overall optimum from stored data _________
QUESTION: E. How do the steps in the questions (a, b, and c) above relate to the steps of this strategy/technique? Note that the question may not completely fulfill the requirements of the approach step, but it does provide some of the work. Not all steps may be addressed by any of these questions.

Dynamic Programming:
- define subproblems/characterize structure
  - and show it is optimal __A,B__
- recursively define the optimum __C__
- compute the optimum storing solutions to subproblems (usually bottom-up) _________
- construct overall optimum from stored data _________
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) ⇒ (I’ll do really well on the exam)

**QUESTION:** What does the implication truth table look like?
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) ⇒ (I’ll do really well on the exam)

**QUESTION:** What does the implication truth table look like?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ⇒ B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) \(\Rightarrow\) (I’ll do really well on the exam)

**QUESTION:** What can you say about the statement if you are told that (I ate plenty of dark chocolate before coming to class) is true?

- a. the statement is true
- b. the statement is false
- c. we cannot say whether the statement is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) $\Rightarrow$ (I’ll do really well on the exam)

**QUESTION:** What can you say about the statement if you are told that (I ate plenty of dark chocolate before coming to class) is true?

a. the statement is true  
b. the statement is false  
c. we cannot say whether the statement is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) ⇒ (I’ll do really well on the exam)

**QUESTION:** What can you say about the statement if you are told that (I’ll do really well on the exam) is true?

a. the statement is true  
b. the statement is false  
c. we cannot say whether the statement is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) \(\Rightarrow\) (I’ll do really well on the exam)

**QUESTION:** What can you say about the statement if you are told that (I’ll do really well on the exam) is true?

- a. the statement is true
- b. the statement is false
- c. we cannot say whether the statement is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) $\Rightarrow$ (I’ll do really well on the exam)

**QUESTION:** What can you say about eating dark chocolate if you are told that the statement is false?

a. (I ate plenty of dark chocolate before coming to class) is true
b. (I ate plenty of dark chocolate before coming to class) is false
c. we cannot say whether (I ate plenty of dark chocolate before coming to class) is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) \(\Rightarrow\) (I’ll do really well on the exam)

**QUESTION:** What can you say about eating dark chocolate if you are told that the statement is false?

- (I ate plenty of dark chocolate before coming to class) is true
- (I ate plenty of dark chocolate before coming to class) is false
- we cannot say whether (I ate plenty of dark chocolate before coming to class) is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) $\Rightarrow$ (I’ll do really well on the exam)

**QUESTION:** What can you say about doing well on the test if you are told that the statement is false?

a. (I’ll do really well on the exam) is true
b. (I’ll do really well on the exam) is false
c. we cannot say whether (I’ll do really well on the exam) is true or false
Assume we have the following statement: (I ate plenty of dark chocolate before coming to class) \(\Rightarrow\) (I’ll do really well on the exam)

**QUESTION:** What can you say about doing well on the test if you are told that the statement is false?

a. (I’ll do really well on the exam) is true
b. (I’ll do really well on the exam) is false
c. we cannot say whether (I’ll do really well on the exam) is true or false
Use the ratio of \( f(n)/g(n) \) to determine the proper relation between the functions. Fill in the blanks with the proper bound using the letters: \( O \), \( \Omega \), or \( \Theta \). The following facts may be useful:
\[
x^y/x^z = x^{y-z} \quad \text{and} \quad x^y x^z = x^{y+z}
\]
\[
c^n/d^n = \left(\frac{c}{d}\right)^n
\]

**QUESTION:** If \( f(n) = 2^n \) and \( g(n) = 3^n \), then
\[
f(n)/g(n) = \ldots , \] which means that
\[
f(n) = \ldots \quad (g(n))
\]
Use the ratio of f(n)/g(n) to determine the proper relation between the functions. Fill in the blanks with the proper bound using the letters: \( \Theta \), \( \Omega \), or \( \Theta \). The following facts may be useful:

\[
\frac{x^y}{x^z} = x^{y-z} \text{ and } x^y x^z = x^{y+z}
\]

\[
\frac{c^n}{d^n} = \left(\frac{c}{d}\right)^n
\]

**QUESTION:** If \( f(n) = 2^n \) and \( g(n) = 3^n \), then

\[
f(n)/g(n) = (2/3)^n
\]

which means that

\[
f(n) = O(g(n))
\]
Use the ratio of \( f(n)/g(n) \) to determine the proper relation between the functions. Fill in the blanks with the proper bound using the letters: \( \mathcal{O}, \Omega, \) or \( \Theta \). The following facts may be useful:

\[
x^y/x^z = x^{y-z} \quad \text{and} \quad x^y x^z = x^{y+z}
\]
\[
c^n/d^n = (c/d)^n
\]

**QUESTION:** If \( f(n) = n^2 \) and \( g(n) = (\sqrt{n})^4 \) then

\[
f(n)/g(n) = \underline{\hspace{2cm}}, \quad \text{which means that}\n\]
\[
f(n) = \underline{\hspace{2cm}} (g(n))
\]
Use the ratio of $f(n)/g(n)$ to determine the proper relation between the functions. Fill in the blanks with the proper bound using the letters: $O$, $\Omega$, or $\Theta$. The following facts may be useful:

\[ x^y/x^z = x^{y-z} \] and \[ x^y x^z = x^{y+z} \]

\[ \frac{c^n}{d^n} = \left(\frac{c}{d}\right)^n \]

**QUESTION:** If $f(n) = n^2$ and $g(n) = (\sqrt{n})^4$ then

\[ f(n)/g(n) = 1, \] which means that

\[ f(n) = \Theta(g(n)) \]
The “simpler” version of the master theorem made an assumption about the terms involved in a recurrence equation.

**QUESTION:** Based on that assumption, is it possible to solve

$$T(n) = 2T(n/2) + \log n$$

using the simple version of the master theorem?

a. yes
b. no
The “simpler” version of the master theorem made an assumption about the terms involved in a recurrence equation.

**QUESTION:** Based on that assumption, is it possible to solve

\[ T(n) = 2T(n/2) + \log n \]

using the simple version of the master theorem?

a. yes  
b. no
QUESTION: What is the solution for:

\[ T(n) = 2T(n/2) + \log n \]
QUESTION: What is the solution for:

\[ T(n) = 2T(n/2) + \log n \]

\[ \Theta(n) \]
QUESTION: What is the solution to the recurrence equation

\[ T(n) = 3T(n/9) + \log n \]
QUESTION: What is the solution to the recurrence equation
\[ T(n) = 3T(n/9) + \log n \]?

\[ \Theta(\sqrt{n}) \]
Assume we have the character frequencies given in the table below:

<table>
<thead>
<tr>
<th>char</th>
<th>P</th>
<th>Q</th>
<th>T</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>freq</td>
<td>17</td>
<td>3</td>
<td>14</td>
<td>20</td>
</tr>
</tbody>
</table>

**QUESTION:** How many bits are in the code for Q? _________
Assume we have the character frequencies given in the table below:

<table>
<thead>
<tr>
<th>char</th>
<th>P</th>
<th>Q</th>
<th>T</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>freq</td>
<td>17</td>
<td>3</td>
<td>14</td>
<td>20</td>
</tr>
</tbody>
</table>

**QUESTION:** How many bits are in the code for Q? __3____
QUESTION: How do you test if x is between 3 and 7 inclusive?
QUESTION: How do you test if x is between 3 and 7 inclusive?

\[ 3 \leq x \leq 7 \]
QUESTION: What is the result of the following commands?

```python
>>> Z = ['h', 'e', 'l', 'l', 'o', 5, 6, 9, 12]
>>> Z[5:2:-1] __________________
```
QUESTION: What is the result of the following commands?

```python
>>> Z = ['h', 'e', 'l', 'l', 'o', 5, 6, 9, 12]
>>> Z[5:2:-1]  
[5, 'o', 'l']
```
QUESTION: What is the result of the following commands?

```python
>>> Z = ['h', 'e', 'l', 'l', 'o', 5, 6, 9, 12]
>>> [::4] ______________________
```

 ANSWER: The result is 'h ll o'.

```python
>>> Z = ['h', 'e', 'l', 'l', 'o', 5, 6, 9, 12]
>>> [::4] 'h ll o'
```
QUESTION: What is the result of the following commands?

```python
>>> Z = ['h', 'e', 'l', 'l', 'o', 5, 6, 9, 12]
>>> [:4] ['h', 'o', 12]
```
Assume the following preferences:

- $m_1: \{w_1, w_2\}$
- $m_2: \{w_2, w_1\}$
- $w_1: \{m_2, m_1\}$
- $w_2: \{m_1, m_2\}$

**QUESTION:** How many stable matchings are there? __________
Assume the following preferences:

\[ m_1: \{w_1, w_2\} \quad w_1: \{m_2, m_1\} \]
\[ m_2: \{w_2, w_1\} \quad w_2: \{m_1, m_2\} \]

**QUESTION**: How many stable matchings are there? \(\boxed{2}\)

\[ \{(m_1, w_1), (m_2, w_2)\} \]
\[ \{(m_2, w_1), (m_1, w_2)\} \]
Here are several logic statements used in a contradiction proof to show that G-S always produces a stable matching:
A. Therefore there is no instability and the set returned by every execution G-S is a stable matching.
B. The set returned by an execution of G-S is never a stable matching.
C. A set S returned by some execution of G-S is not a stable matching.
D. In S, there exist pairs (m, w) and (m’, w’) such that m prefers w’ to w and w’ prefers m to m’.
E. In S, there exist pairs (m, w) and (m’, w’) such that m prefers w’ to w and w prefers m’ to m.
F. w’ rejected m, and traded up to higher and higher men in her list, culminating in m’.

**QUESTION:** What order should these statements be put into to complete the proof? ______________________
Here are several logic statements used in a contradiction proof to show that G-S always produces a stable matching:
A. Therefore there is no instability and the set returned by every execution G-S is a stable matching.
B. The set returned by an execution of G-S is never a stable matching.
C. A set S returned by some execution of G-S is not a stable matching.
D. In S, there exist pairs \((m,w)\) and \((m',w')\) such that \(m\) prefers \(w'\) to \(w\) and \(w'\) prefers \(m\) to \(m'\).
E. In S, there exist pairs \((m,w)\) and \((m', w')\) such that \(m\) prefers \(w'\) to \(w\) and \(w\) prefers \(m'\) to \(m\).
F. \(w'\) rejected \(m\), and traded up to higher and higher men in her list, culminating in \(m'\).

**QUESTION:** What order should these statements be put into to complete the proof? \(\text{C, D, F, A}\)