22F-COM SCI-35L-LEC-1 Midterm

[redacted]

TOTAL POINTS

93.5 / 100

QUESTION 1

1 1a 6 / 6

+ **0 pts** Incorrect

✓ + **3 pts** Hard links cannot point to directories

✓ + **3 pts** Symbolic links can be set to any string and/or can point to directories

QUESTION 2

2 1b 8 / 8

✓ + **4 pts** First part: if you hard link a relative symlink S to a different location, then the file referred to may be different than S.

+ **1 pts** First part: Hard link points symlink, which is different than S itself. Not what the question was asking but shows some mastery.

✓ + **4 pts** Second part: Hard links point to the same inode. Always map to the same file.

+ **0 pts** Incorrect

QUESTION 3

3 2a 5 / 5

✓ - **0 pts** Correct

- **2.5 pts** No or wrong explanation

- **2.5 pts** Didn’t mention or Wrong output (never mentioned that it will output nothing)

- **1 pts** Minor bugs in explanations

- **1 pts** Minor bugs in output (eg. mention it will output nothing but also say it will outputs an error. there won’t be any errors but only output nothing)

QUESTION 4

4 2b 5 / 5

✓ - **0 pts** Correct

- **5 pts** Wrong description of the output and explanation

- **2.5 pts** Wrong understanding of the output (Expecting answers like "No permission to run or execute 'mystery' anymore")

- **2.5 pts** Wrong explanation (Expecting answers related to file permission analysis)

- **3.5 pts** Moderate try.

QUESTION 5

5 2c 5 / 5

✓ - **0 pts** Correct

- **2 pts** no or wrong explanation of why "./" is needed (expecting sth like "looking up in the current dir for execution")

- **1.5 pts** No or wrong description of what will happen if "./" is omitted;

- **1 pts** Insufficient explanation of "what will happen if "./" is omitted" (need to mention it will look for mystery in system env variable $PATH)

- **1.5 pts** No or wrong explanation of the last question "might it still work anyway?" Insufficient explanation (What if the current working dir is in $PATH?)

QUESTION 6

6 2d 5 / 5

✓ - **0 pts** Correct

- **2.5 pts** Wrong output (Expecting answers like "no output" or "nothing")

- **2.5 pts** Wrong or No explanation (Expecting answers related to the "RegEx requires at least one alpha or ":" ")

QUESTION 7

7 2e 6 / 6

+ **1 pts** shows some understanding

+ **2 pts** recognize that ":" must be present

+ **2 pts** recognize the hierparts are all optional because of the ? sign

+ **2 pts** recognize $scheme can match any letter (upper or lower)

+ **0 pts** wrong

✓ + **6 pts** correct

QUESTION 8

8 2f **6 / 8**

- **0 pts** understand there're double evaluation of shell and grep, and also understand that . is not a special character in shell script so single \ is just as fine

✓ - **2 pts** **understand that shell script would be evaluated first then the evaluated results will be what grep will take.**

But . is actually not a special character in shell script so single \ would work just fine because . would not be escaped, and since \ is not used to escape, \ would be preserved to grep, so \ and \\ work just same.

- **6 pts** understand \ or . is a special character, but did not understand that the special characters are evaluated first by the shell script, and then the evaluated texts would be again evaluated by grep. so \\ will be evaluted to \ after shell, and then evaluate to escape . in grep.

- **8 pts** incorrect

QUESTION 9

9 3a **5 / 6**

- **6 pts** Blank

- **0 pts** Correct

- **1 pts** Only remove line 13.

✓ - **1 pts** **Didn't remove related documentation line.**

- **5 pts** Incorrect change.

- **3 pts** Delete/change too many lines.

- **0.5 pts** No need to call (push-mark) after the changes.

- **4 pts** Didn't remove the code part.

+ **1 pts** deleted related documentation lines.

- **0.5 pts** didn't completely remove line11-13

QUESTION 10

10 3b **6 / 8**

- **0 pts** Correct

- **8 pts** blank

- **7 pts** Wrong answer

- **4 pts** Incorrect go-to position

- **1 pts** syntax error

- **1 pts** incorrect arg check

✓ - **1 pts** **should be arg - 1**

- **2 pts** lack code

- **3 pts** should forward-line instead of goto-char

✓ - **1 pts** **use forward-line instead of goto-line in code.**

- **1 pts** No default action.

QUESTION 11

11 4a **6 / 6**

✓ + **1 pts** **2-4 Random not imported**

✓ + **1 pts** **7 random imported so getting output index from random.choice**

✓ + **1 pts** **9 getting output index from random.choice for string**

✓ + **3 pts** **11-15 Line 377, 241, and 242 returns 0 as index that is out of range**

+ **1.5 pts** 11-15 partial explanation (something related to code implementation)

+ **0 pts** No answer

QUESTION 12

12 4b **8 / 8**

✓ + **4 pts** **Correct a: worst case time complexity will be infinity**

+ **2 pts** a Partially correct

✓ + **4 pts** **Correct b: < 2 microseconds (explained)**

+ **2 pts** Partially Correct b: < 1 microseconds / < 2 microsecond partially incorrect explanation

+ **1 pts** incorrect b: Partially correct explanation

+ **4 pts** Partially correct explanation for both parts

+ **2 pts** Something relevant

+ **0 pts** Incorrect

QUESTION 13

13 5 **8 / 8**

+ 0 pts incorrect / no answer

+ 2 pts incorrect but shows some understanding

+ 2 pts did not explain enough

+ 4 pts did not demonstrate full understanding

+ 6 pts did not demonstrate full understanding

✓ **+ 8 pts** Correct

QUESTION 14

14 6 **6 / 6**

✓ **- 0 pts** Correct

- 6 pts blank

- 2 pts No explanation on how to debug python, e.g. print values, use pdb.

- 2 pts No explanation on how to debug JS.

- 1 pts Error in python debugging.

- 1 pts Error in JS debugging.

- 5 pts Can't recognize the writing

QUESTION 15

15 7 **8.5 / 10**

- 0 pts Correct

- 2 pts Element Structure partially incorrect

- 3 pts Attributes incorrect

- 1.5 pts Attributes partially incorrect

- 3 pts Text incorrect

✓ **- 1.5 pts** **Text alignment partially incorrect (Should have multiple T under small)**

- 1 pts Minor mistakes (In either of above three)

- 8 pts Partially correct

- 9 pts Something done but incorrect

- 10 pts No answer

UCLA CS 35L midterm, fall 2022
100 minutes total, open book, open notes, closed computer.
1 minute = 1 point. Write answers on exam.

Name: [redacted] Student ID: [redacted]

1a (6 minutes). Explain why the shell command 'ln . dot' fails while 'ln -s . dot' succeeds.

. is a link referring to the current directory ($PWD). ln -s . dot succeeds as it is permissible to create a soft link to another link and or to a directory.

ln . dot fails as you are attempting to create a hard link to a directory, which is not allowed.

1b (8 minutes). Explain why a hard link H to a symbolic link S can refer to a different file than S does, whereas a hard link H to another hard link J must reference the same file that J references.

Suppose S is soft link to file F1 and J is a hard link to file F2.

Hard links point to the same inode as the file they link to. Soft links merely point to another file. Since hardlinks point to inodes, they are absolube. J refers to the same inode as F2 and as such, H refers to the same inode as J & F2. If we have the following directory structure: (-> is soft and => is hard link)

~/tmp1: ~/tmp2:
S->F1 H=>~/tmp1/S
F1 F1

Then S refers to ~/tmp1/F1 but H refers to ~/tmp2/F1

2. Suppose the file 'mystery' is mode 755 and contains the following contents, derived from a simplified version of Internet RFC 3986, "Uniform Resource Identifier (URI): Generic Syntax":

```
#!/bin/sh
alpha='a-zA-Z';  digit='0-9';  unreserved="[$alpha$digit._~-]"
scheme="([$alpha][$alpha$digit+.-]*)"
hexdig="[A-Fa-f$digit]";  pct_encoded="(%$hexdig$hexdig)"
sub_delims='[!$&'\''()*+,;=]'
userinfo="(($unreserved|$pct_encoded|$sub_delims|:)*)"
deco="([$digit]|[1-9][$digit]|1[$digit][$digit]|2[0-4][$digit]|25[0-5])"
IPv4address="($deco\\.$deco\\.$deco\\.$deco)"
reg_name="(($unreserved|$pct_encoded|$sub_delims)*)"
host="($IPv4address|$reg_name)";  port="([$digit]*)"
authority="(($userinfo@)?$host(:$port)?)"
pchar="($unreserved|$pct_encoded|$sub_delims|:|@)"
segment="($pchar*)";  segment_nz="($pchar+)"
path_abempty="((/$segment)*)"
path_absolute="(/($segment_nz(/$segment)*)?)"
path_rootless="($segment_nz(/$segment)*)"
hierpart="((//$authority$path_abempty|$path_absolute|$path_rootless)?)"
query="(($pchar|/|\\?)*)"
fragment="($query)"
grep -E "$scheme:$hierpart(\\?$query)?(#$fragment)?"
```

a:

2a (5 minutes). What is the output of the following shell command? Briefly explain.

```
./mystery </dev/null
```

Nothing as you are effectively running: grep -E <stuff> /dev/null. It doesn't matter what <stuff> is as /dev/null contains nothing so grep will match nothing

2b (5 minutes). Suppose 'mystery' were mode 644 instead of 755. How would that affect the output of the shell command in (a)?

110·100·100

Still nothing as the file has no execute permissions, so nobody, not even owner, can execute it. An error would be thrown.

2c (5 minutes). Briefly explain why the './' is needed before 'mystery' in the shell command in (a). What would happen if './' were omitted? Might the command still work anyway?

./ is needed as we can assume mystery is not contained in a folder specified by $PATH. If mystery were in a folder specified in $PATH (such as /usr/bin), then we could run it without the ./ If not, however it would tell you that the "mystery command is not found.

2d (5 minutes). What is the output of the shell command "echo '' | ./mystery"? Briefly explain.

Nothing as we can tell without even looking at variable values that we need at least a ":" in the string to match.

2e (6 minutes): What is a shortest nonempty string FOO such that the shell command "echo 'FOO' | ./mystery" outputs something? Explain.

Any "[a-zA-Z]:" would match (a:, Z:, etc).

This is evident as hierpart, query, and fragment all contain ? quantifiers which permit nothing to be matched. scheme, however needs an alpha followed by some optional text (so just an alpha). The final string also requires a :, hence my answer

2f (8 minutes). Almost every occurrence of '\' in 'mystery' is doubled, so it appears as '\\'. Explain why this doubling is needed. Explain what would happen if every instance of '\\' were replaced by a single '\', for example, in the IPv4address line.

\ is used to escape characters. Here, \\ is used to escape the \ character so it can be represented as a literal \ This is needed as many strings are substituted into other strings, so the escaping happens twice. Without this, IPv4address, for example, would escape the "\." to just "." Then when matching IPv4address later, grep would see "." and try and match any character, rather than "\." which merely matches a literal "."

3. You're annoyed that the following Emacs keystrokes:

```
  C-u 2 M-<
```

put you 20% (i.e., 2/10) of the way through the current buffer. You'd rather have that command put you on line 2 of the current buffer instead. You type the following

```
  C-h k M-<
```

and see the following on your screen:

```
  M-< runs the command beginning-of-buffer (found in global-map), which
  is an interactive byte-compiled Lisp function in 'simple.el'.

  It is bound to <begin>, C-<home> and M-<.  It can also be invoked
  from the menu: Edit → Go To → Goto Beginning of Buffer

  (beginning-of-buffer &optional ARG)

  Move point to the beginning of the buffer.
  With numeric arg N, put point N/10 of the way from the beginning.
  If the buffer is narrowed, this command uses the beginning of the
  accessible part of the buffer.

  Push mark at previous position, unless either a C-u prefix is
  supplied, or Transient Mark mode is enabled and the mark is active.
```

This documentation bothers you in a different way, because you think this is too complicated, and that the last sentence quoted above should be removed and the resulting code fixed accordingly.

[continued on next page]

You click on 'simple.el' in the above text, and see the following function definition:

```
      1  (defun beginning-of-buffer (&optional arg)
      2    "Move point to the beginning of the buffer.
      3  With numeric arg N, put point N/10 of the way from the beginning.
      4  If the buffer is narrowed, this command uses the beginning of the
      5  accessible part of the buffer.
      6
      7  Push mark at previous position, unless either a \\[universal-argument
] prefix
      8  is supplied, or Transient Mark mode is enabled and the mark is active
."
      9    (declare (interactive-only "use `(goto-char (point-min))' instead."
))
     10    (interactive "^P")
     11    (or (consp arg)
     12        (region-active-p)
     13        (push-mark))
     14    (let ((size (- (point-max) (point-min))))
     15      (goto-char (if (and arg (not (consp arg)))
     16                     (+ (point-min) 1
     17                        (/ (* size (prefix-numeric-value arg)) 10))
     18                   (point-min))))
     19    (if (and arg (not (consp arg))) (forward-line 1)))
```

if Arg:

else:

3a (6 minutes). Change the definition of beginning-of-buffer so that it no longer implements the bothersome last sentence of the documentation. Use the line numbers of the source code in your description of the changes you'd make.

Delete lines 11, 12, and 13. These are the only lines that deal with the conditional pushing of the mark.

3b (8 minutes). Further change the definition of beginning-of-buffer so that with numeric arg N it puts you on line N rather than N/10 of the way through the buffer.

- delete line 19, this is not needed after the following new changes.
- replace lines 15-18 with:

```
(if (and arg (not ((consp arg)))
    (goto-line (prefix-numeric-value arg))
    (goto-char (point-min))
)
```

Consider the following source code adapted and simplified from random.py:

```
239     def _randbelow(self, n):
240         "Return a random int in the range [0,n).  Return 0 if n==0."
241         if not n:
242             return 0
243         getrandbits = self.getrandbits
244         k = n.bit_length()  # don't use (n-1) here, as n can be 1
245         r = getrandbits(k)  # 0 <= r < 2**k
246         while r >= n:
247             r = getrandbits(k)
248         return r
...
375     def choice(self, seq):
376         """Choose a random element from a non-empty sequence."""
377         return seq[self._randbelow(len(seq))]
...
826     def getrandbits(self, k):
827         "getrandbits(k) -> x.  Generates an int with k random bits."
...
```

n ≤ 2^k-1

and consider the following python3 transcript on SEASnet:

```
 1  >>> random.choice([1,2,3])
 2  Traceback (most recent call last):
 3    File "<stdin>", line 1, in <module>
 4  NameError: name 'random' is not defined
 5  >>> import random
 6  >>> random.choice([1,2,3])
 7  2
 8  >>> random.choice("abcdef")
 9  'a'
10  >>> random.choice("")
11  Traceback (most recent call last):
12    File "<stdin>", line 1, in <module>
13    File "/usr/local/cs/Python-3.10.7/lib/python3.10/random.py", line 3
77, in choice
14      return seq[self._randbelow(len(seq))]
15  IndexError: string index out of range
```

n

15

1010

16 2^n

$0 \le r \le 16$

16

10000

4a (6 minutes). Explain Python’s output in lines 2-4, 7, 9, and 11-15.

2-4: random has not been imported, so the python shell has not access tho that module's namespace.

7: random.choices invokes _randbelow to return a random element from a sequence. 2 is in the given list, so in this case 2 happens to be returned.

9: In this case the logic is the same as line 7 but the sequence is a string and the randomly returned element is the string "a".

11-15: Here an error is thrown on line 377. len(seq) returns 0 as it is an empty string and thus self._randbelow(len(seq)) also returns 0. Then, whe indexing seq[0], an error is thrown as you can't get the 0th index of an empty sequence

4b (8 minutes). n.bit_length() returns the number of bits needed to represent n in binary, excluding the sign and leading zeros. Assuming that getrandbits and bit_length each finish in less than one microsecond, that all the other code takes zero time, and that ‘choice’ is called on a sequence of length 1,000,000, how much CPU time will the ‘choice’ method take, in the worst case? Also, how would your answer change if ‘choice’ is called on a sequence of length 1,048,576 (2**20) instead? Briefly justify your answer.

244: <1us

245: <1us

246-247:

In theory, because the conditional in the while loop relies on random chance, this code could run forever in the worst case, as getrandbits could keep returning numbers greater than 1,000,000 or greater than 1,048,576. Only a sequence of length 2^n-1 would have a finite worst case run-time of <2us, as the conditional in the while loop is guaranteed to be false first try.

5 (8 minutes). HTTP/3 is often touted as a major performance win for web applications. Explain why your solution to Assignment 3 would not materially benefit from switching to HTTP/3.

HTTP/3 is built atop UDP. UDP relies on single, shorter messages being sent to avoid missing packet delays. This is useful for more high preformance webapps, but assignment 3 is rather simple and runs (mostly) locally. This means that an improved networt protocol would have little imact on Assignment 3 preformance

6 (6 minutes). Suppose you were supposed to use Python as much as possible when doing assignment 3, thus minimizing the use of JavaScript. Explain how this would affect how you'd develop your solution to Assignment 3. In particular, how would you debug your program, compared to debugging your program with Node.js as it is?

Development would require a lot more backend program programming as the only practical use of python would be for a backend. While node.js would handle displaying all game info, python would handle the logic of player movement and player status. Debugging would occur mostly in an external editor or command line instead of the browser, but similar debugging practices could be used as js and py are both interpreted, scripting languages.

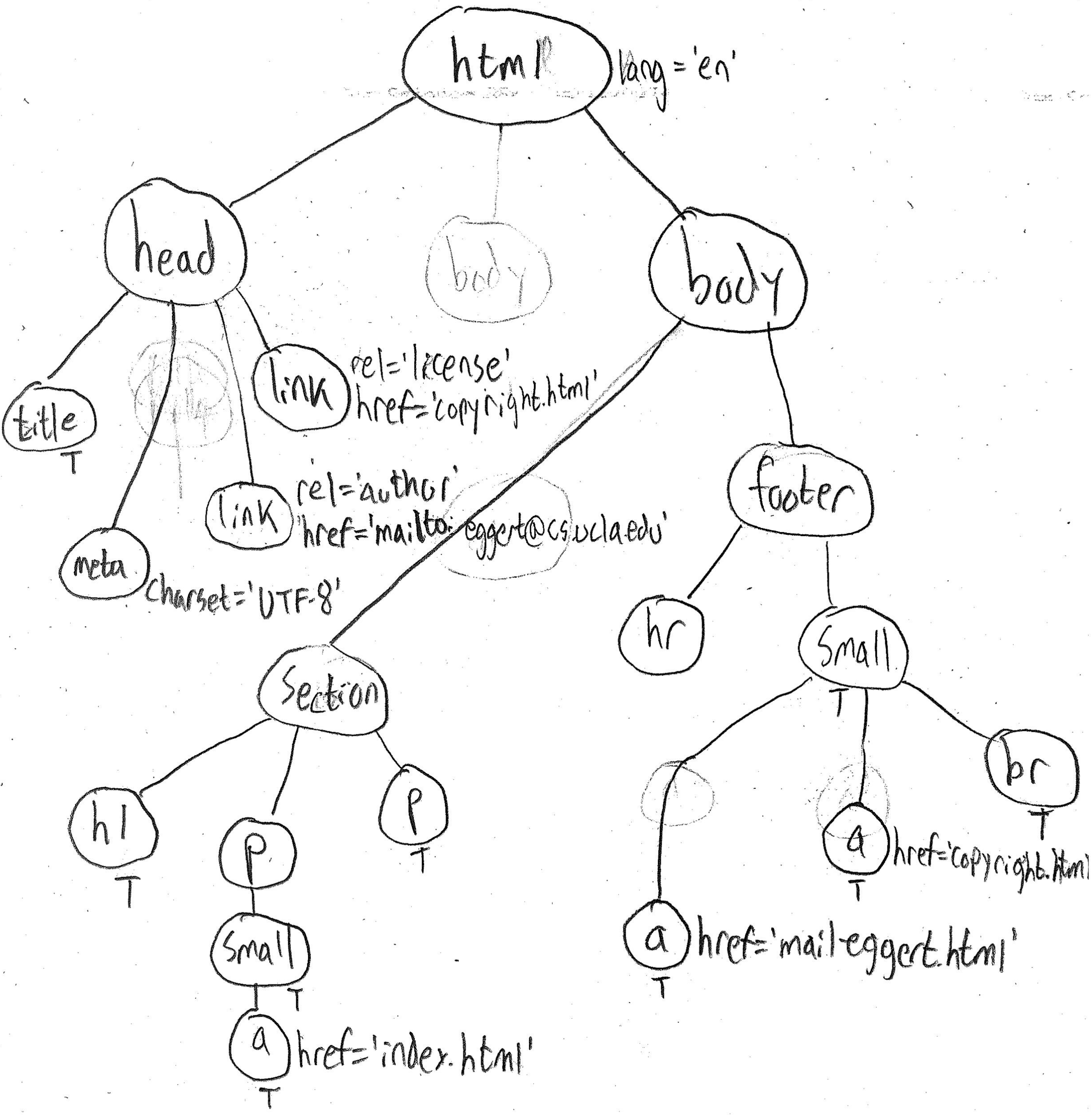
7 (10 minutes). Diagram the DOM tree for the following HTML5 document. (You need not diagram the first line.) Assume that elements lacking closing tags are void.

```
<!DOCTYPE html>
<html lang='en'>
<head>
<title>Assignments for UCLA Computer Science 35L, fall 2022</title>
<meta charset='UTF-8'>
<link rel='author' href='mailto:eggert@cs.ucla.edu'>
<link rel='license' href='copyright.html'>
</head>
<body>
<section>
<h1>Assignments for UCLA Computer Science 35L, fall 2022</h1>
<p><small>[<a href='index.html'>35L home</a>]</small></p>
<p>Most assignments are divided into two parts: laboratory exercises and
  homeworks. Laboratory exercises are expected to be done with more
  assistance from the TAs and LAs.</p>
</section>
<footer>
<hr>
<small>
 &copy; 1999, 2003&ndash;2022 <a href='mail-eggert.html'>Paul Eggert</a>.
 See <a href='copyright.html'>copying rules</a>.<br>
 $Id: assign.html,v 1.186 2022/10/13 01:13:20 eggert Exp $
</small>
</footer>
</body>
</html>
```

In your diagram, circle the element nodes. Represent each text string by a capital “T” under the element node, at the appropriate position. (For example, write a “T” to represent “Assignments for UCLA Computer Science 35L, fall 2022”.) Write “A='V'” to the right of each node that has an attribute A with value V (using the actual attribute name and value, not a literal capital “A” or capital “V”).

[Put the diagram on the next page.]

[Diagram the DOM tree described in the previous page. Neatness counts.]


html
lang='en'
head
body
title
T
link
rel='license'
href='copyright.html'
meta
Charset='UTF-8'
link
rel='author'
href='mailto: eggert@cs.ucla.edu'
footer
hr
Small
T
Section
h1
T
p
p
T
small
T
a
href='index.html'
T
a
href='mail-eggert.html'
T
a
href='copyright.html'
T
br
T