University of Waterloo
CS240 Fall 2017
Assignment 5

Written Questions Due Date: Wednesday, November 29, at 5:00pm
Programming Question Due Date: Monday, November 4, at 5:00pm

Problem 1

a)

c	a	b	c	d	o	t
L(c)	7	3	-1	5	6	-1

i	0	1	2	3	4	5	6	7
S[i]	-7	-6	-5	-4	-3	-2	-1	6

b)

d	o	t	a	d	o	t	a	d	o	t	d	o	t	a	d	o	b	o	d	o	a	d	o	t
						X	a																	
													X	a										
														a	d	o	b	o	d	o	a			

c) Let $k = \lfloor n/m \rfloor$ and $r = n mod m$ (so $n = km + r$). Consider pattern a^m and the text $((a^{m-1}b)^k a^r$. Boyer-Moore will first compare the last character of P with the first b of T, find a mismatch and shift forward with the bad character shift. But b does not occur in P so it will shift P to start at the character after the first b. The next comparison would then be with the last character of P with the second b of T. This repeats, so Boyer-Moore exactly compares all the bs of T. At the last b, it will shift P out of range and stop having done exactly $k = \lfloor n/m \rfloor$ comparisons.

d) For any $m \geq 1$ and any $n \geq m$ that is a multiple of m, give a pattern P and a text T such that the Boyer-Moore algorithm looks at all characters of the text at least once and returns with failure. Justify your answer.

Consider the pattern $P = ca^{m-1}$ and the text $(ba^{m-1})^k$ where $k = m/n$. Since Boyer-Moore starts at the end of the patter and works backwards, it looks at the first copy of (ba^{m-1}) and compares all a characters as successful matches and fails on the comparison with b. So it has looked at all characters of this part of the text. It then shifts forward using the suffix-skip all the way past the initial (ba^{m-1}) to align with the next copy of (ba^{m-1}). This sequence then repeats for each copy of (ba^{m-1}), hence all characters of the text are looked at.

e) A number of heuristics can be used with Boyer-Moore to reduce the number of comparisons performed between P and T. Suppose we use Boyer-Moore with only the Peek heuristic. The Peek heuristic states that if $P[j] \neq T[i]$ and $P[j-1] \neq T[i-1]$ then the next location to search for P at is $T[i+m-1]$. Show that the Peek heuristic may fail to find P in T, i.e., find a pattern P, and a text T containing P, such that Peek fails to find P in T.

Suppose Peek does guarantee that P will be found in T, then Peek should find $P = alpha$ in $T = abcalphabana$. P mismatches T at indices 4 and 3 then shifts the pattern too far so the initial a of P aligns with the l of $T[4]$. This shift is too far forward so P is not found in T. However, P is found in the T starting at index 3.

Problem 2

a) $h(123) = 6$. The entry in each column of the table is the hash of the next three digits.

7	9	2	3	6	5	7	4	0	1	6	5	2	4	1	0	6	9	3	1	7	8	3	0	1	2	3
8	4	1	4	8	6	1	5	7	2	3	1	7	5	7	5	8	3	1	6	8	1	4	3	6		

We successfully find 123 at the last non-blank entry. At the other entries with a 6 (these are the two false positives), we need to do a string comparison to determine whether our entry is correct.

b) $h(123) = 11$. Once again, the entry in each column of the table is the hash of the next three digits.

7	9	2	3	6	5	7	4	0	1	6	5	2	4	1	0	6	9	3	1	7	8	3	0	1	2	3
48	43	20	29	41	38	36	17	8	21	36	28	17	18	10	21	45	43	21	26	47	38	13	4	11		

Since there are no hash conflicts, the only string comparison happens at the 11 near the end, at which point the pattern is found. Thus, there are 0 false positives.

Problem 3 Suffix Trie [4+4=8marks]

a) Draw the suffix tree for $T = deacacaeacacaedd$.

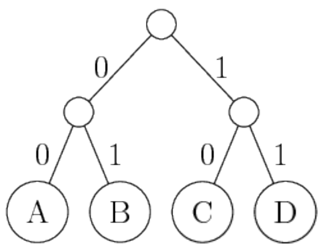
b) Trace a search for $P = aca$ in the suffix trie created in the previous part.

Problem 4

a) The trie is created by combining: B with D; then A with (BD); and finally (ABD) with C.
A has code 00
B has code 010
C has code 1
D has code 011

$$
\begin{aligned}
\mathrm{WPL}(T) &= f(A)d(A) + f(B)d(B) + f(C)d(C) + f(D)d(D) \\
&= 3 \cdot 2 + 2 \cdot 3 + 3 \cdot 1 + 1 \cdot 3 \\
&= 18
\end{aligned}
$$

b)



$$
\begin{aligned}
\mathrm{WPL}(T') &= f(A)d(A) + f(B)d(B) + f(C)d(C) + f(D)d(D) \\
&= 3 \cdot 2 + 2 \cdot 2 + 3 \cdot 2 + 1 \cdot 2 \\
&= 18
\end{aligned}
$$

This tree cannot be built by Huffman because Huffman requires B and D to be merged first, whereas in this example either A, B or C, D were merged first.

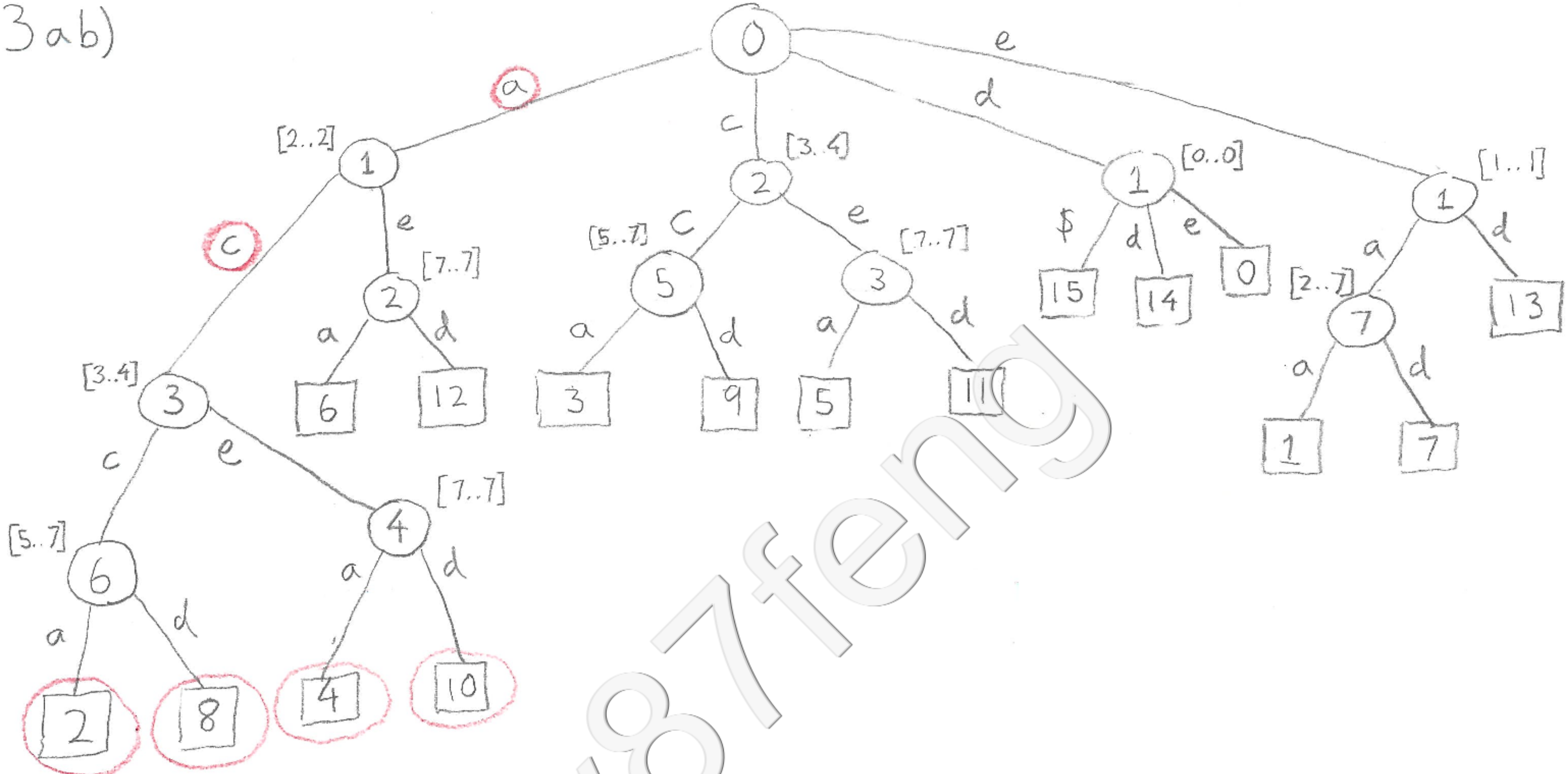
c) Suppose $f_i > f_j$ for two characters c_i, c_j in a Huffman tree. Then $d_i \leq d_j$.

Proof by contradiction: $d_i > d_j$ for some Huffman tree T and recall that the Huffman tree minimizes the WPL. Then let T' be the tree obtained by exchanging the nodes c_i, c_j. Then:

$$
\begin{aligned}
\mathrm{WPL}(T) &= \sum_{k=1}^{n} f_k \cdot d_k \\
&= \sum_{k \neq i,j} f_k \cdot d_k + f_i \cdot d_i + f_i \cdot d_k \\
&= \mathrm{WPL}(T') + f_i \cdot d_i + f_j \cdot d_j - f_i \cdot d_j - f_j \cdot d_i \\
&= \mathrm{WPL}(T') + f_i(d_i - d_j) - f_j(d_i - d_j) \\
&= \mathrm{WPL}(T') + \underbrace{(f_i - f_j)}_{\geq 0}\underbrace{(d_i - d_j)}_{\geq 0} \\
&\geq \mathrm{WPL}(T')
\end{aligned}
$$

So T is not optimal and cannot be a Huffman tree, a contradiction. So if $f_i > f_j$ then $d_i \leq d_j$ as desired.

3ab)


0
a
c
d
e
[2..2]
1
[3..4]
2
[0..0]
1
[1..1]
1
c
e
[7..7]
2
[5..7]
5
[7..7]
3
$
d
e
15
14
0
a
d
[2..7]
7
13
[3..4]
3
a
d
6
12
a
d
3
9
a
d
5
11
a
d
1
7
c
e
[7..7]
4
[5..7]
6
a
d
2
8
a
d
4
10