ECS 122b Midterm 2

Geoffrey Mohn

TOTAL POINTS

64.5 / 85

QUESTION 1

1 Problem 1 8 / 10

- **- 2 pts** Minor mistake in recurrence relation
- **- 3 pts** Major mistake in recurrence relation
- **- 2 pts** Wrong Worst case Sl/Sr
- **- 3 pts** Final answer not arrived at
- **- 0 pts** Correct

✓ **- 2 pts** *No steps given(master's theorem)*

- **- 4 pts** Wrong split for your y
- **- 10 pts** Not Attempted/completely wrong

Good job, but you haven't shown that you arrive at O(n) using master's theorem.

QUESTION 2

2 Problem 2 12.5 / 15

- **- 0 pts** Correct

✓ **- 2.5 pts** *Did not choose y for worst case analysis*

- **- 5 pts** Either summation to calculate nodes/edges diverges or the result is exponential runtime
- **- 5 pts** Result is O(nlogn) or O(logn). Should be O(n)
- **- 15 pts** No analysis/Blank Answer/No real attempt at answer
- **- 5 pts** Used y = 2 as in lecture rather than regular y
- **- 5 pts** No shown summation but tree structure present
- **- 5 pts** Forgot to multiply result by n
- **- 5 pts** Summation equation is incorrect/or no summation
- **- 7.5 pts** Only defines a tree of size 13
- **- 15 pts** not work shown/ critical error

QUESTION 3

3 Problem 3 (a) 10 / 10

✓ **- 0 pts** *Correct*

- **- 3 pts** Order of result has an error
- **- 5 pts** No discernable linear time algorithm
- **- 4 pts** No statement of SA

QUESTION 4

4 Problem 4 17 / 20

- **- 0 pts** Correct
- **- 10 pts** Part(a) wrong/not attempted
- **- 10 pts** Part (b) wrong

✓ **- 3 pts** *Not used the z algo for Part (a)*

- **- 4 pts** Major mistake in suffix array runtime
- **- 2 pts** Minor mistake in suffix array runtime
- **- 4 pts** Major mistake in suffix tree runtime
- **- 2 pts** Minor mistake in suffix tree runtime
- **- 1 pts** Wrong final result for Part (b)
- **- 2 pts** Runtime not given for part (a)

Good job but you haven't used z algo for part (a)

QUESTION 5

5 Problem 3(b),c,d 17 / 30

- **0 pts** Correct

✓ **- 7 pts** *no work for LCP*

✓ **- 6 pts** *not linear BWT*

- **4 pts** bwt is not indexes its a charcters in a string, this is a large mistake as there would be no way to reverse back from numbers nor could we encode and compress.

- **1 pts** Minor mistake in final BWT

- **3 pts** Major mistake in final BWT

- **5 pts** Not encoded suffix tree for (c)

- **5 pts** Wrong final BWT

- **10 pts** Wrong SA and LCP

- **5 pts** Mistake in LCP only

- **4 pts** Major Mistakes in encoded suffix tree

- **2 pts** Minor mistakes in encoded suffix tree

Name: Gaffrey Cuohn
STUDENT ID: 912568148
Midterm 2

You will use **your y** variable which is calculated by taking just the single last digit of your student **id modding it by 3** and then **adding 8** to it. Everything in the test is self-explanatory. If you are confused, state your assumption and answer the question. We will grade with your assumption in mind. **We cannot answer any questions related to your interpretation.**

y calculation example: For example, my last digit is 3. Hence **3%3=0**. Therefore **y=0+8=8**, I will use the number 8 in place of y everywhere on the test.

What is your y ? 10

Do not forget to fill in your y in the problems below. Do not solve with y being a variable. **Fill in your constant value. Show all work and calculations, you do not need to simplify.**

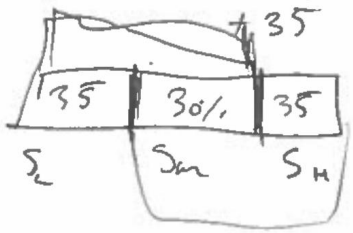
Problem 1. 10pts
In class, we analyzed randomized selection by looking at the expected number of times you select a pivot before the pivot you select is within the middle 50% percentile. Please analyze randomized selection given that it takes **a-times** to select a pivot before it is within **(20+y)%.**

$T(n) = (a-1)O(n) + O(n) + T\left(\frac{65}{100}n\right)$

$O(n)$

$\sum_{n=0} \frac{65}{100^i}$

$(20+10) = 30\%$



y = 10

Problem 2. 15pts

Given a tree with n leaves and that each internal node must have either y or y+1 children, what is the tightest upper bound on the number of nodes in the tree? What about edges?

10 or 11 children

#of nodes = 11n

$E = 11n - 1$

Problem 3. 40pts

gungungun

Let string M=121212. Replace all 1's with the first letter of your first name and 2's with the first letter of your last name. For me, this would lead to M=yfyfyf. Please make sure you do this correctly because we will grade you down if you do not. You will use the new M string going forward.

What is your string M: gungungun$ (1 2 3 4 5 6 7)

a.) Calculate SA of M in linear time, showing only the top level steps of the recursion like in class.

$SA_{1,2} = [7, 5, 1, 4, 2]$

$SA_3 = [3, 6]$

$SA_{1,2}[3] = \text{"gungung"} > SA_3[1]$ "gungun$"

- 7) $ 1
- 5) gun$ 2
- 1) gungungun$ 4
- 4) ungun$ 6
- 2) ungungun$ 7
- 3) gungun$ 3
- 6) un$ 5

$LCP(Suff_{K+1}, Suff_{j+1}) = h-1$

b.) Calculate LCP of M in linear time.

c.) Draw the *first four lexicographically smallest* suffixes on an encoded suffix tree.

d.) Create the BWT for M in linear time.

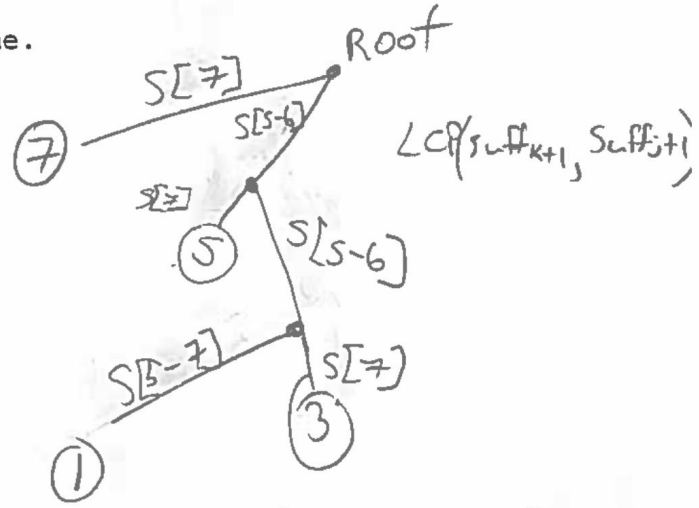
$SA = [7, 5, 3, 1, 6, 4, 2]$

$LCP[0, 2, 4, 0, 0, 3]$

1 $gungungun
5 un$gungu
2 gun$gungun
6 ngun$gung
3 gungun$gun
7 ungungun$g
4 gungungun$

→ Sort →

u_1
u_2
u_3
$
g_1
g_2
g_3



Problem 4. 20pts

a.) Provide the most efficient pseudocode for searching for the reverse of a pattern P inside a circular string T, using Z-algorithm. Given |P|=n and |T|=n^6. Analyze the run-time of the algorithm.

$S = T.P$

$T = Splice(T[rand])$

```
for i = |T| to 1; i--
    if P == T[i-|P|]
        Z[i-|P|] = |P|
```

$O(P+T) = n + n^6 \rightarrow O(n^6)$

b.) Given you are going to search for n^7 patterns each approximately length n inside a text of length n^14. Compare which algorithm is more asymptotically efficient suffix trees or suffix arrays.

$T = n^{14}$ $\quad x = n^7$ $\quad P = n$ $\quad K = depth?$

$ST = O(x(P+K))$

$SA = O(x(P \log T))$

$ST = O(n^7(n + n^7)) = O(n^{14})$

$SA = O(n^7(n \log n^{14})) = O(n^8 \log n^{21})$

Use Suffix Tree