

CS251 – Homework 4: String I

Due: April 15, 2016

Dylan Mackey

1. B) n

Since the pattern is the reverse of S, the length is also n. Therefore, you only need n comparisons because as soon as one character does not match, it is false, but you do need to compare every character using the brute force algorithm.

2. First, find the length of the string. Then compare the first character with the last, followed by the second character to the second to last, etc. Once you reach the middle of the string, you are done with $n/2$ comparisons.

3. A) True. The lengths have to be equal

B) False. You need to compare the end of the S2 string, not the beginning

C) True. This will compare opposite characters

D) False. J's condition should be $j = m - i - 1$, not $j = m - i + 1$. This will cause a segmentation fault

4. Alice is correct. For example: "stressed" and "dessert". If you only compare $n/2$ characters, Bob's method would seem to be correct. However, he is not because dessert does not have the final 's'.

5.

C	t	e	s	d
L(c)	3	4	2	5

*Note: All other characters in the alphabet would have an index of -1.

6. t e s t i n g a t e s t e d t e s t t e x t

t e s t e d 1 (n is not in the string, so skip 6 places)

t e s t e d 1 (then shift 2 places to match up the t's)

t e s t e d 6 (match)

8 Comparisons

7. A) $O(nm+s)$ Standard Boyer-Moore Algorithm

B) You can delete the bottom else section, where it assigned i, l, and j. This is because semordnilaps have to be exactly equal, so if you find one mismatch, you can return false. This is $O(n)$ where n is the pattern length.

8. G A C A G A T G A

J	0	1	2	3	4	5	6	7	8
P(J)	G	A	C	A	G	A	T	G	A
F(J)	0	0	0	0	1	2	0	1	2

GA is the longest available substring to skip to.

9. G G T A C C C G A C A G A T G A C A G A

G A C A G A T G A 1

G A C A G A T G A 1

G A C A G A T G A 1

G A C A G A T G A 1

G A C A G A T G A 1

G A C A G A T G A 1

G A C A G A T G A 1

G A C A G A T G A 9

No jumps were able to be made, so it performed like a brute force algorithm

16 Comparisons

10. A) True. Based on the question, you must continue the algorithm to find more
- B) True. While the big O stays the same, the running time as a whole will increase as the size of the text increases.
- C) False. A match must be all of the words of the pattern, so a substring on a match cannot be a match. However, another match can **start or end** within a previous match.
- D) False. Once the algorithm goes through the text, it should find all of the matches.

11. "she sells sea shells"

Output	Index	string
(0,s)	1	s
(0,h)	2	h
(0,e)	3	e
(0, ' ')	4	' '
(1,e)	5	se
(0,l)	6	l
(6,s)	7	ls
(4,s)	8	" s"
(3,a)	9	ea
(8,h)	10	" sh"
(3,l)	11	el
(7,)	12	ls

12.

Output	Index	Sting
(0,f)	1	F
(0,l)	2	L
(0,a)	3	A
(0,s)	4	S
(0,h)	5	H
(0, ' ')	6	' '
(0, ' ')	7	' '
(1, l)	8	Fl
(3, s)	9	As
(5, ' ')	10	"h "
(0, -)	11	-
(7, h)	12	" h"
(0, u)	13	U
(0, n)	14	N
(0, d)	15	D
(0, r)	16	R
(0, e)	17	E
(15, ' ')	18	"d "
(0, y)	19	Y
(3, r)	20	Ar
(18, d)	21	"d d"
(9, h)	22	ash
(0, ")	23	

Flash, flash – hundred yard dash