

Announcements:

1. HW 11 is due Thursday. It is a very long assignment; that is why I gave you a whole week for it.
Today at noon is the halfway point between the due dates for HW 10 and HW 11. **Are you about halfway done with HW11?**
2. **No class meeting Thursday.** I will be in my office during class time to assist you with HW11 or HW 12.
3. **B-trees** (section 7.4 in Levitin) should be straightforward for those who have had experience with other balanced trees, so I am asking you to read this section on your own and ask questions about anything you do not understand.
4. I will be off-campus Oct 30 in the afternoon and most of Oct 31 (I hope to be here for hours 9-10) due to my IVIG infusions. **No class meeting Oct 31.**
5. **Exam 2** Tuesday Nov 4 in class.
6. **In my office today:** hours 6, 8, 10.

Main ideas from today:

1. Why is the “ $-k$ ” in the formula for Boyer-Moore bad-symbol shift?
 $d_1 = \max\{t_1(c) - k, 1\}$, where $t_1(c)$ is the value from the Horspool shift table.

2. **Boyer-Moore Algorithm:** After successfully matching $0 < k < m$ characters, with a mismatch after k matches from the end of the pattern (the corresponding mismatched character in the text is c), the algorithm shifts the pattern right by
$$d = \max \{d_1, d_2\}$$
where $d_1 = \max\{t_1(c) - k, 1\}$ is the bad-symbol shift. $t_1(c)$ is the entry for c from the Horspool table.
 $d_2(k)$ is the good-suffix shift

3. (4 points) With one or two other students, try to come up with rules for creating the good shift table for a pattern string of length m . **Input:** the pattern string. **Output:** a table of $m-1$ shift values. $gs[k]$ is the amount that we can shift the pattern if the last k characters of the pattern match the text. [domain: $k = 1..m-1$]
Example patterns to help you think about this: CABABA, AWOWWOW, WOWWOW, ABRACADABRA.

4. For each given string, fill in the good-suffix table from the Boyer-Moore algorithm. Once again, work with one or two other students.

1. banana

k	shift
1	
2	
3	
4	
5	

2. wowwow

k	shift
1	
2	
3	
4	
5	

3. abcdcbcabcbc

k	shift
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	