University of Waterloo
CS240E, Winter 2025
Written Assignment 5 Post-Mortem

This document goes over common errors and general performance on the assignment. We create it using feedback from the graders, and it is meant to be used as a resource to understand common areas that we can improve in.

Question 1 [3+3=6 marks]

- Part (a) was done well.
- In part (b), some solutions argued along the lines of the Boyer-Moore pseudocode and last-occurrence array. It is possible to do this correctly, but some solutions incorrectly concluded that the shift amount is always m. This can't be true for every text T, so be sure to "sanity-check" your answers in cases like these.

Question 2 [4 marks]

- This question was done well.

Question 3 [2+2+6=10 marks]

- Parts (a) and (b) were done well.
- For part (c), some solutions stated that the Huffman trie for S would be a full binary tree without justifying why rigourously.

Question 4 -

Question 5 [2+2+3+2+2+5+2=18 marks]

- Parts (a) and (b)(i) were done well.
- In part (b)(ii), some solutions didn't account for decoding the first $m-1$ characters of T_2 before searching through the indices of T_1 for P.
- In part (b)(iii), many solutions made the subtle error of forgetting that one character from T_2 is added to $\mathcal{W}$. Handling this case required using the assumption that $|P| > 1$.
- Part (b)(iv) was done well, with many solutions offering humourous remarks about the size of their constants.

- In part (b)(v), some solutions did not fully explain why the number of tuples was bounded by a constant, or why $P_2, \ldots, P_{k-1}$ are uniquely determined by P_1.
- In part (b)(vi), a few solutions forgot to use parts (i) and (ii) to handle the case where the pattern overlaps T_1.

Question 6 [2+2+5=9 marks]

- In part (a), some solutions had an off-by-one error in the suffix array, perhaps due to misremembering the definition. This was not penalized, but be sure to get the indices right on the final!
- Part (b) was done well.
- In part (c), many solutions tried to maintain a set of suffixes of T that are candidates for a prefix match with P. While it is possible to achieve this efficiently, most of these solutions had a runtime of $\Omega(mN)$.