Homework 2

Michael Morikawa

April 28, 2020

Chapter 11

R-11.8 Suppose we are given two n-element sorted sequences A and B that should not be viewed as sets (that is, A and B may contain duplicate entries). Describe an $O(n)$-time method for computing a sequence representing the set $A \cup B$ (with no duplicates).

Answer:

First we will have indexes i and j starting at 0. In order to ignore duplicates within the 2 sequences check to see if A[i] == A[i+1] and shift until they are not equal or i+1 is out of bounds and do the same for B but using j instead. Then compare A[i] and B[j] and put the smaller one into the new set and increment the correct index, i for A and j for B. If they are equal take from A and increment both i and j.

R-11.23 Describe, in pseudo-code, how to perform path compression on a path of length h in O(h) time in a tree-based partition union/find structure.

Answer:

```
while p is not root:
    prev = p
    p = p->parent
    prev->parent = root
```

R-11.25 Describe an in-place version of the quick-select algorithm in pseudo-code.

Answer:

Algorithm: inPlaceQuickSelect(S, a, b, k):

Input: Array S, int a, b, k

Output: k^{th} smallest element in S

```
if a ≥ b: return
if k>0 and k< b - a + 1:
    p = S[b] (pivot)
    l = a
    r = b - 1
    while l ≤ r:
        while l ≤ r and S[l] ≤ p:
            l++
        while r ≥ l and S[r] ≥ p:
            r++
        if l < r:
            swap S[l] and S[r]
swap S[l] and S[b]
index = l (set index to pivot location)
if index -a == k - 1:
    return S[index]
(Recur on left side)
if index - a is larger than k - 1:
    return inPlaceQuickSelect(S, a,index -1, k)
(Recur on right side and adjust k)
return inPlaceQuickSelect(S, index + 1, r, k - index + l - 1)
```

C-11.3 Given two sets A and B represented as sorted sequences, describe an efficient algorithm for computing $A \oplus B$, which is the set of elements that are in A or B, but not in both.

Answer:

We can use the generic merge class from the book and create a class SymmetricDifferenceMerge that inherits from it. In that class we overload the fromA and fromB functions to push the inPlaceQuickSelect into the ouput set and for fromBoth we ignore it and do nothing. This will discard elements that are in both sets.

Chapter 12

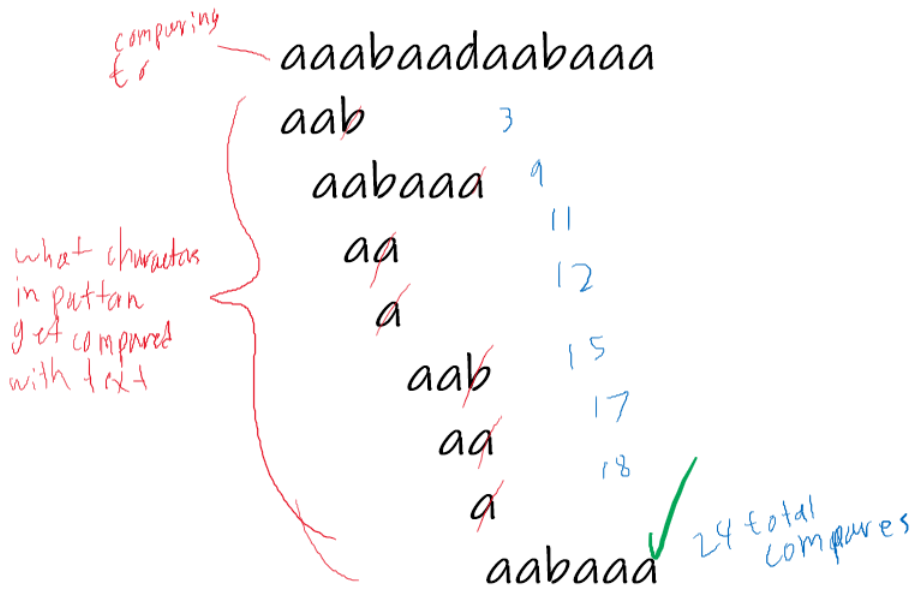
R-12.3 List the prefixes of the string P ="aaabbaaa" that are also suffixes of P.

Answer:

"", "a", "aa", "aaa"

R-12.4 Draw a figure illustrating the comparisons done by brute-force pattern matching for the text "aaabaadaabaaa" and pattern "aabaaa".

Answer:

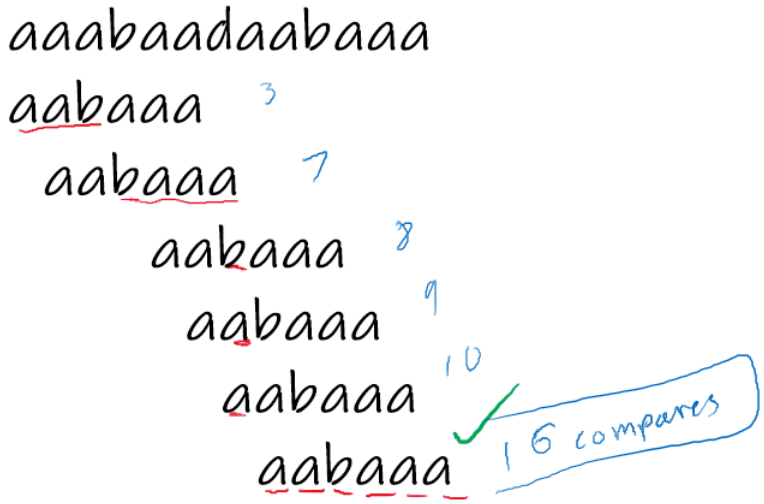


R-12.5 Repeat the previous problem for the BM pattern matching algorithm, not counting the comparisons made to compute the last(c) function.

Answer:

R-12.6 Repeat the previous problem for the KMP pattern matching algorithm, not counting the comparisons made to compute the failure function.

Answer:



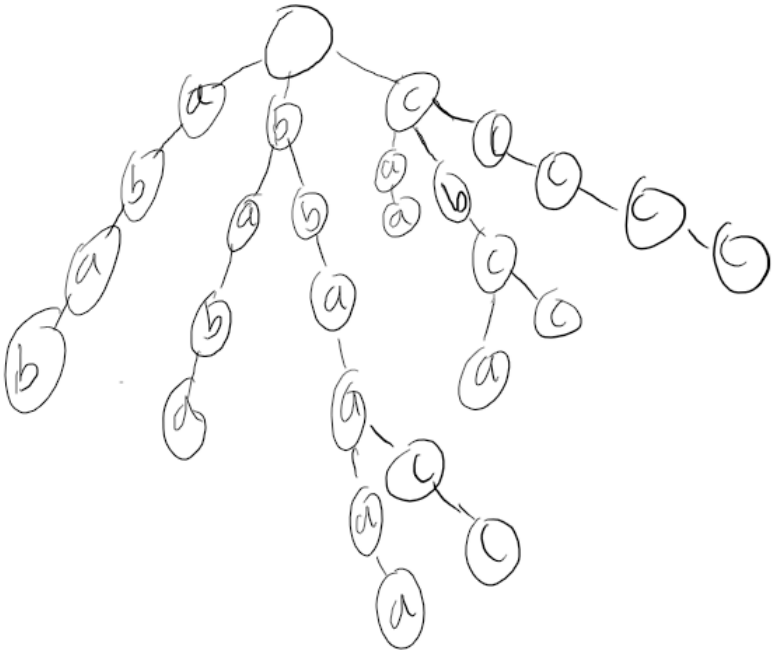
R-12.9 Compute a table representing the KMP failure function for the pattern string "cgtacgttcgtac".

Answer:

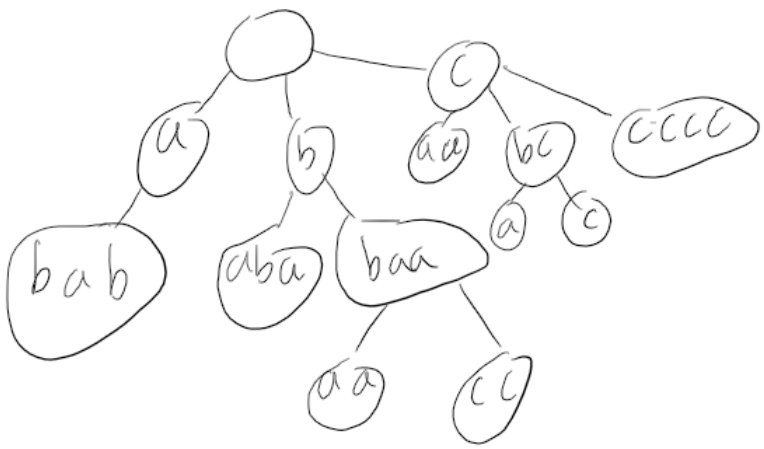
c	g	t	a	c	g	t	t	c	g	t	a	c
0	0	0	0	1	2	3	0	1	2	3	4	5

R-12.10 Draw a standard trie for the following set of strings:
{abab, baba, ccccc, bbaaaa, caa, bbaacc, cbcc, cbca}.

Answer:



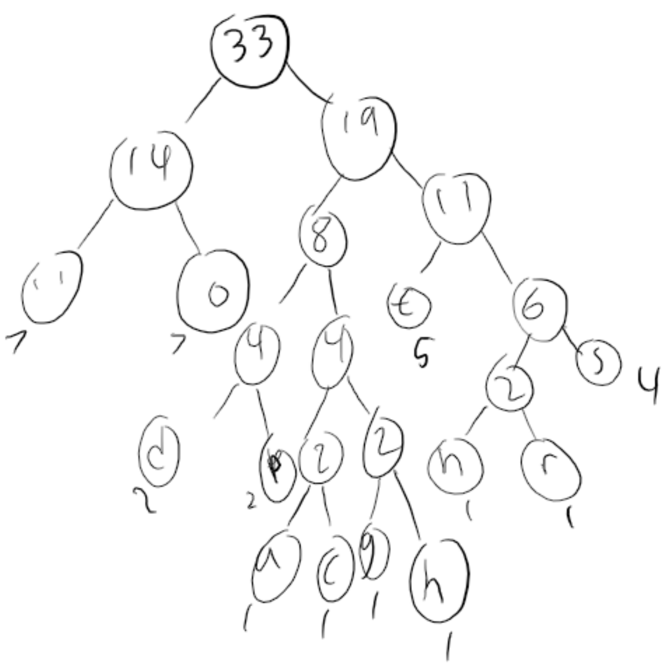
R-12.11 Draw a compressed trie for the set of strings given in Exercise R-12.10.
Answer:



R-12.13 What is the longest prefix of the string "cgtacgttcgtacg" that is also a suffix of this string?
Answer:
cgtacg

R-12.14 Draw the frequency array and Huffman tree for the following string:
"dogs do not spot hot pots or cats".
Answer:

Character		a	c	d	g	h	n	o	p	r	s	t
Frequency	7	1	1	2	1	1	1	7	2	1	4	5



C-12.1 A native Australian named Anatjari wishes to cross a desert carrying only a single water bottle. He has a map that marks all the watering holes along the way. Assuming he can walk k miles on one bottle of water, design an efficient algorithm for determining where Anatjari should refill his bottle in order to make as few stops as possible. Argue why your algorithm is correct.
Answer:

Use a greedy algorithm that chooses the watering hole that is the maximal distance still less than k miles. Do so until he reaches the end.

C-12.3 Give an example set of denominations of coins so that a greedy change- making algorithm will not use the minimum number of coins.

Answer:

25, 10, 1 to make 55 would use25, 25, 1, 1, 1, 1, 1 but the optimal solution is 25, 10, 10, 10.