Homework Assignment #6

Due Time/Date

2:20PM Tuesday, November 1, 2022. To be better prepared for the midterm exam on October 25, you are advised to complete it before the exam. Late submission will be penalized by 20% for each working day overdue.

How to Submit

Please use a word processor or scan hand-written answers to produce a single PDF file. Name your file according to this pattern: "b107050xx-hw6". Upload the PDF file to the NTU COOL site for Algorithms 2022. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

There are five problems in this assignment, each accounting for 20 points. (Note: problems marked with "(X.XX)" are taken from [Manber 1989] with probable adaptation.)

1. Consider the solutions to the union-find problem discussed in class. Suppose we start with a collection of ten elements: A, B, C, D, E, F, G, H, I, and J.

   (a) Assuming the balancing, but not path compression, technique is used, draw a diagram showing the grouping of these ten elements after the following operations (in the order listed) are completed:

      i. union(A,B)
      ii. union(E,F)
      iii. union(A,F)
      iv. union(C,D)
      v. union(A,C)
      vi. union(G,H)
      vii. union(C,G)
      viii. union(A,D)
      ix. union(I,J)
      x. union(A,I)

   In the case of combining two groups of the same size, please always point the second group to the first.

   (b) Repeat the above, but with both balancing and path compression.

2. Below is the pseudocode of the binary search algorithm we discussed in class. Would the code still be correct if we change the assignment "$Middle := \lceil \frac{Left+Right}{2} \rceil$" to "$Middle := \lfloor \frac{Left+Right}{2} \rfloor$" for *Middle* to take instead the largest integer less than or equal to $\frac{Left+Right}{2}$? Please justify your answer. If the modified code is incorrect, what other changes must be made accordingly?

function Find $(z, Left, Right) : integer$;
begin
  if $Left = Right$ **then**
    if $X[Left] = z$ **then** $Find := Left$
    else $Find := 0$
  else
    $Middle := \lceil \frac{Left+Right}{2} \rceil$;
    if $z < X[Middle]$ **then**
      $Find := Find(z, Left, Middle - 1)$
    else
      $Find := Find(z, Middle, Right)$
end

Algorithm Binary_Search (X, n, z);
begin
  $Position := Find(z, 1, n)$;
end

3. (6.32) Prove that the sum of the heights of all nodes in a complete binary tree with n nodes is at most $n - 1$. (A complete binary tree with n nodes is one that can be compactly represented by an array A of size n, where the root is stored in $A[1]$ and the left and the right children of $A[i]$, $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$, are stored respectively in $A[2i]$ and $A[2i + 1]$. Notice that, in Manber's book a complete binary tree is referred to as a balanced binary tree and a full binary tree as a complete binary tree. Manber's definitions seem to be less frequently used. Do not let the different names confuse you. "Balanced binary tree" in the original problem description is the same as "complete binary tree")

4. Consider the *next* table as in the KMP algorithm for string $B[1..9] = abaababaa$.

1	2	3	4	5	6	7	8	9
a	b	a	a	b	a	b	a	a
-1	0	0	1	1	2	3	2	3

Suppose that, during an execution of the KMP algorithm, $B[6]$ (which is an a) is being compared with a letter in A, say $A[i]$, which is not an a and so the matching fails. The algorithm will next try to compare $B[next[6] + 1]$, i.e., $B[3]$ which is also an a, with $A[i]$. The matching is bound to fail for the same reason. This comparison could have been avoided, as we know from B itself that $B[6]$ equals $B[3]$ and, if

$B[6]$ does not match $A[i]$, then $B[3]$ certainly will not, either. $B[5]$, $B[8]$, and $B[9]$ all have the same problem, but $B[7]$ does not.

Please adapt the computation of the *next* table so that such wasted comparisons can be avoided. Also, please give the values of the new *next* table for the string $B[1..9] = abaababaa$, according to the adaptation.

5. (6.17) Given two strings $A = ababa$ and $B = bbbaba$, what is the result of the minimal cost matrix $C[0..5, 0..6]$, according to the algorithm discussed in class for changing A character by character into B? Aside from giving the cost matrix, please show the details of how the entry $C[3, 4]$ is computed from the values of $C[2, 3]$, $C[2, 4]$, and $C[3, 3]$.