Assignment 6

Yu-Chieh Kuo B07611039[†]

[†]Department of Information Management, National Taiwan University

Problem 1

I do not draw diagrams by `tikz` for convenience and save time since I am sort of busy, so I draw them by iPad and display them as Fig. 1 and 2.

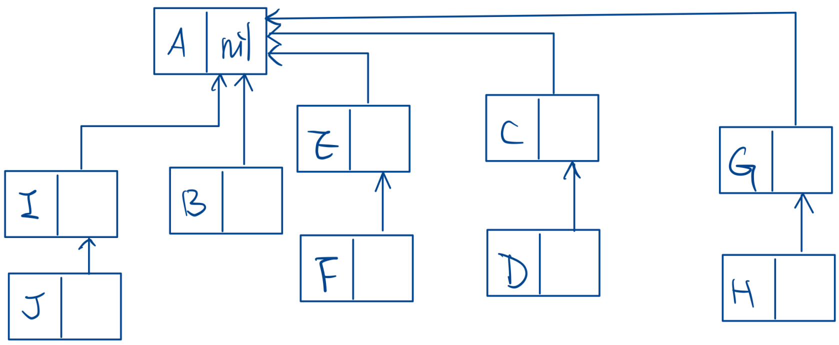

Figure 1: The union diagram.

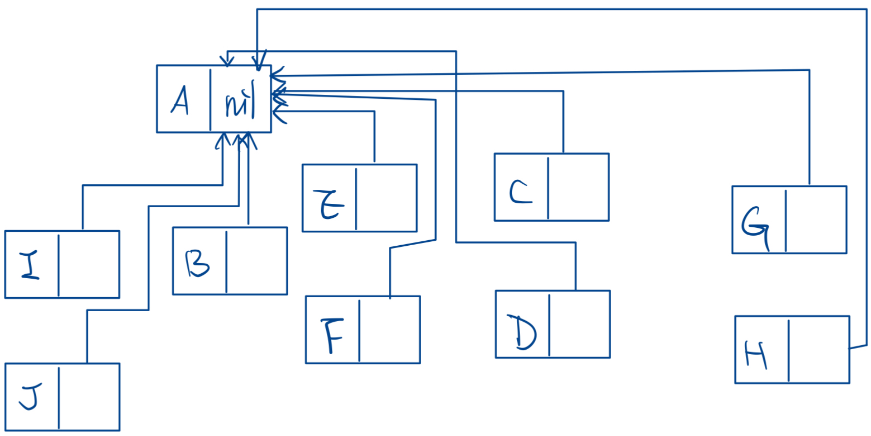

Figure 2: The union diagram after balancing and path compression.

Problem 2

The code would be incorrect, if just that change is made. Consider an array with two numbers 7 and 9, says $X[7, 9]$. Suppose we implement the binary search to find out whether 6 is in X. The execution will set *Middle* to $\lfloor \frac{Left+Right}{2} \rfloor = 1$ Since $6 < 7 = X[1] = X[Middle]$, the algorithm invokes $Find(6, 1, 0)$, which will result in an access to $X[0]$, an erroreous behavior.

To modify the the code, it can be add an `if` statement before invoking the *Find* function (See line 10 to 14 in Algorithm 1).

Algorithm 1 Corrected Binary Search

```
1: function CorrectedBinarySearch(z, Left, Right)
2:     if Left = Right then
3:         if X[Left] = z then
4:             Find := Left
5:         else
6:             Find := 0
7:         end if
8:     else
9:         Middle := ⌈(Left+Right)/2⌉
10:        if z < X[Middle] then
11:            if Middle ≤ 0 then
12:                Stop the algorithm and print not found
13:            else
14:                Find := Find(z, Left, Middle − 1)
15:            end if
16:        else
17:            Find := Find(z, Middle, Right)
18:        end if
19:    end if
20: end function
```

Problem 3

We have already known that the sum of the heights of all nodes in a full binary tree of height h is $2^{h+1} - h - 2$. Let $G(n)$ denote the sum of the heights of all nodes in a complete binary tree with n nodes. For a full binary tree (a special case of complete binary trees) with $n = 2^{h+1} - 1$ nodes where h is the height of the tree, we already know that $G(n) = 2^{h+1} - (h+2) = n - (h+1) \leq n - 1$. Given this knowledge, we prove the general case of arbitrary complete binary trees by induction.

For base case ($n = \{1, 2\}$): When $n = 1$, the tree is the smallest full binary tree with one single node whose height is 0. Hence, $G(n) = 0 \leq 1 - 1 = n - 1$. When $n = 2$,the tree has one additional node as the left child of the root. The height of the root is 1, while that of its left child is 0. Consequently, $G(n) = 1 \leq 2 - 1 = n - 1$.

For inductive step ($n > 2$): If n happens to be equal to $2^{h+1}1$ for some $h \geq 1$, *i.e.* the tree is full, then we are done. Note that this covers the case of $n = 3 = 2^{1+1} - 1$. On the other hand, suppose $2^{h+1} - 1 < n < 2^{h+2} - 1 (h \geq 1)$ *i.e.* the tree is a proper complete binary tree with height

$h+1 \geq 2$. We observe that at least one of the two subtrees of the root is full, while the other is complete (possibly full). There are three cases remaining to consider and discuss.

1. The left subtree is full with n_1 nodes and the right one is complete but not full with n_2 nodes (such that $n_1 + n_2 + 1 = n$). In this case, both subtrees much be of height h and $n_1 = 2^{h+1} - 1$. From the special case of full binary trees and the induction hypothesis, $G(n_1) = 2^{h+1} - (h+2) = n_1 - (h+1)$ and $G(n_2) \leq n_2 - 1$. $G(n) = G(n_1) + G(n_2) + (h+1) \leq (n_1 - (h+1)) + (n_2 - 1) + (h+1) = (n_1 + n_2 + 1) - 2 \leq n - 1$.

2. The left subtree is full with n_1 nodes and the right one is also full with n_2 nodes. In this case, the left subtree much be of height h and $n_1 = 2^{h+1} - 1$, while the right subtree much be of height $h-1$ and $n_2 = 2^h - 1$. From the special case of full binary trees, $G(n_1) = 2^{h+1} - (h+2) = n_1 - (h+1)$ and $G(n_2) = 2^h - (h+1) = n_2 - h$. $G(n) = G(n_1) + G(n_2) + (h+1) \leq (n_1 - (h+1)) + (n_2 - h) + (h+1) = (n_1 + n_2 + 1) - (h+1) \leq n - 1$.

3. The left subtree is complete but not full with n_1 nodes and the right one is full with n_2 nodes. In this case, the left subtree much be of height h, while the right subtree much be of height $h-1$ and $n_2 = 2^h - 1$. From the induction hypothesis and the special case of full binary trees, $G(n_1) \leq n_1 - 1$ and $G(n_2) = 2^h - (h+1) = n_2 - h$. $G(n) = G(n_1) + G(n_2) + (h+1) \leq (n_1 - 1) + (n_2 - h) + (h+1) = (n_1 + n_2 + 1) - 1 = n - 1$.

Problem 4

Algorithm 2 Modified KMP

```
function ModifiedKMP(B,m)
    for i := 3 ~ m do
        j := next[i]
        while j > 0 & B[i] = B[j + 1] do
            j := next[j + 1]
        end while
        next[i] := j
    end for
end function
```

The corresponding consequences from Algorithm 2 alter to $\{-1, 0, 0, 1, 0, 0, 3, 0, 1\}$.

Problem 5

The cost matrix is computed as

$$C[3,4] = \min \left\{ \begin{array}{l} C[2,4] + 1 \\ C[3,3] + 1 \\ C[2,3] \end{array} \right\} = 2.$$