**CSCI 3333**

**Data Structures**

**Summer 2008**

**Suggested Solution to Mid-Term Examination**

(1)

(a) T

(b) T

(c) F

(d) T

(e) T

(2) For example:

**Algorithm** IsSubset(A, B)

**Input**: A, B: integer arrays of distinct elements

**Output**: whether A is a subset of B, i.e., every element in A appears in B.

If sizeOf(A) > sizeOf(B) return false;

If sizeOf(A) == 0 return true;

S <- Sort(A);

T <- Sort(B);

NS <- SizeOf(S);

NT <- SizeOf(T);

indexT <- 0;

for (int i=0 to NS-1) loop

 while (T[indexT] <S[i]) loop

 if (indexT >= NT-1) return false;

 indexT++;

 end loop;

 if (T[indexT] > S[i]) return false;

end loop;

return true;

(3) Original: {6, 19, 22, 4, 7, 21, 26, 99, 1, 16, 24, 47, 72, 13, 2, 86, 33, 18}

**5-sorted:**

Conceptual View:

Before:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 19 | 22 | 4 | 7 |
| 21 | 26 | 99 | 1 | 16 |
| 24 | 47 | 72 | 13 | 2 |
| 86 | 33 | 18 |  |  |

After:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 19 | 18 | 1 | 2 |
| 21 | 26 | 22 | 4 | 7 |
| 24 | 33 | 72 | 13 | 16 |
| 86 | 47 | 99 |  |  |

Result array: {6, 19, 18, 1, 2, 21, 26, 22, 4, 7, 24, 33, 72, 13, 16, 86, 47, 99}

**3-sorted:**

Conceptual View:

Before:

|  |  |  |
| --- | --- | --- |
| 6 | 19 | 18 |
| 1 | 2 | 21 |
| 26 | 22 | 4 |
| 7 | 24 | 33 |
| 72 | 13 | 16 |
| 86 | 47 | 99 |

After:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 4 |
| 6 | 13 | 16 |
| 7 | 19 | 18 |
| 26 | 22 | 21 |
| 72 | 24 | 33 |
| 86 | 47 | 99 |

Result array: {1, 2, 4, 5, 13, 16, 6, 19, 18, 26, 22, 21, 72, 24, 33, 86, 47, 99}

**1-sorted:**

Result Array: sorted.

(4)

(a) ab+c\*d/e-

(b) --a\*b/cd-ef

(5) For example,

Function CountLarger(p: BST, target: Integer)

Input:

 p: the root reference of a BST.

 target: the target value to compare with.

Output: the number of nodes in p with a key value greater than target

if (p == null) return 0;

if (p.key < target)

return CountLarger(p.right, target);

end if;

if (p.key == target)

 return size(p.right);

else /\* p.key > target \*/

 return CountLarger(p.left, target) + size(p.right) + 1;

end if;

(6)

(a) [[[] 1 2] 4 [[[5 16 []] 23 []] 80 99]]

(b) [10 16 [[17 18 []] 30 77]]

(7)

(a) [ [[a – b] \* [c + d]] / [e – f] ]

(b) O(N2)

(c) 1