

CS2092 Programming Lab

Test 4: Data Abstraction- II

November 3, 2015

Name Roll No. Desk No. Time: 45 Min.

- [0] Assume the existence of the following subfunctions/selectors for Binary Search Trees. (You may use these without defining/coding them):
- (a) **is-empty?** which takes a **bst** as input and returns **#t** or **#f**.
 - (b) **search** which takes a **bst** and a **value** as input and returns the **bst** rooted at the value if present; else returns an **empty-bst**.
 - (c) **get-val** which takes a **bst** as input and returns the **value** of root node.
 - (d) **get-ls** which takes a **bst** as input and returns the **left-subtree**.
 - (e) **get-rs** which takes a **bst** as input and returns the **right-subtree**.
- [1] Given a **bst** and two values **x** and **y**, write a function (**is-descendent? x y bst**) which checks whether **x** is a descendent of **y** in **bst**. You may assume that (**is-descendent? x x bst**) returns **#t**.
- [2] Given a **bst** and two values **x** and **y**, write a function (**get-smallest-subtree x y bst**) to find the smallest subtree containing both **x** and **y**.
In addition to selectors defined above, you **should** only use *is-descendent?* function.
- You may assume that, for both questions, nodes with values **x** and **y** are both present in the given **bst**.
 - We say that **x** is a descendent of **y** if and only if **x** belongs to the subtree of **y**. Further if **x** is a descendent of **y**, then **y** is an ancestor of **x**.
-

(define (is-descendent? x y bst)

)