CISS350: Data Structures and Advanced Algorithms
Quiz q10704

Name: YOUR EMAIL Score:

The following is the node for the binary search tree.

```
class BSTNode
{
public:
    int key_;
    BSTNode * parent_;
    BSTNode * left_, * right_;
};
```

Q1. Complete the function below to return the address of the BST node with the given target key value. If it is not found, return `NULL`. Calling `bst_find(p, 42)` will search for the node with key value 42 among all the children of the node with address `p`. (This is just our usual BST find.)
Answer:

```
BSTNode * bst_find(BSTNode * p, int target)
{
}
```

Q2. Complete the function below to return the address of the *parent* node of the BST node with the given target key value. If it is not found, return `NULL`. (There are two cases where `NULL` is returned: when the target key is at the root of the tree and when the target key is not in the tree.)
Answer:

```
BSTNode * bst_find_parent(BSTNode * p, int target)
{
}
```

Q3. Complete the following function which is similar to the BST find. If the target key value is found, the address of this target node is returned and `found` (pass by reference) is set to `true`. If the target key value is not found, the address of the last node reached is returned and `found` is set to `false`. For instance when you perform this version of find on the following tree starting at `5` and searching for target value `7`

```
     (5)
    /   \
```

```
  (3)     (8)
         /   \
      (6)     (9)
```

the address returned is the address of the node with value `6` and `found` is set to `false`. (This function is helpful: after failing to find a target key value and you want to insert this new key, it tells you where is the parent node for the new node.)
Answer:

```
BSTNode * bst_find_up_to(BSTNode * p, int target, bool & found)
{
}
```

Instructions

In the file `thispreamble.tex` look for

```
\renewcommand\AUTHOR{}
```

and enter your email address:

```
\renewcommand\AUTHOR{jdoe5@cougars.ccis.edu}
```

(This is not really necessary since alex will change that for you when you execute `make`.) In your bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`.

Enter your answers in `main.tex`. In the bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`.

For each question, you'll see boxes for you to fill. For small boxes, if you see

```
1 + 1 = \answerbox{}.
```

you do this:

```
1 + 1 = \answerbox{2}.
```

`answerbox` will also appear in "true/false" and "multiple-choice" questions.

For longer answers that need typewriter font, if you see

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
\end{answercode}
```

you do this:

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
int x;
\end{answercode}
```

`answercode` will appear in questions asking for code, algorithm, and program output. In this case, indentation and spacing is significant. For program output, I do look at spaces and newlines.

For long answers (not in typewriter font) if you see

```
What is the color of the sky?
\begin{answerlong}
\end{answerlong}
```

you can write

```
What is the color of the sky?
\begin{answerlong}
The color of the sky is blue.
\end{answerlong}
```

A question that begins with "T or F or M" requires you to identify whether it is true or false, or meaningless. "Meaningless" means something's wrong with the question and it is not well-defined. Something like "$1 + 2 = 4$" is either true or false (of course it's false). Something like "$1+_2 = 4$?" does not make sense.

When writing results of computations, make sure it's simplified. For instance write 2 instead of $1 + 1$.

Higher level classes.

For students beyond 245: You can put LaTeX commands in `answerlong`.

More examples of meaningless statements: Questions such as "Is $42 = 1+_2$ true or false?" or "Is $42 = \{2\}^{\{3\}}$ true or false?" does not make sense. "Is $P(42) = \{42\}$ true or false?" is meaningless because $P(X)$ is only defined if X is a set. For "Is $1 + 2 + 3$ true or false?", "$1 + 2 + 3$" is well–defined but as a "numerical expression", not as a "proposition", i.e., it cannot be true or false. Therefore "Is $1 + 2 + 3$ true or false?" is also not a well-defined question.

More examples of simplification: When you write down sets, if the answer is $\{1\}$, do not write $\{1, 1\}$. And when the values can be ordered, write the elements of the set in ascending order. When writing polynomials, begin with the highest degree term.

When writing a counterexample, always write the simplest.