

**Homework 4**

Instructors: *Leo Porter* (Sec. A)  
& *Debashis Sahoo* (Sec. B)

**Due on:** Thursday Nov. 15th (40 points)

Name: \_\_\_\_\_ PID: \_\_\_\_\_

**Instructions**

1. Answer each problem in the boxes provided. Any writing outside of the boxes *will NOT be graded*. Do not turn in responses recorded on separate sheets.
  2. Handwritten or typed responses are accepted. In either case, make sure all answers are in the appropriate boxes.
  3. All responses *must* be neat and legible. Illegible answers will result in zero points.
  4. Make sure to *scan in portrait mode* and to *select the corresponding pages* on Gradescope for each question.
  5. You may use code from any of the class resources, including Stepik. You may not use code from other sources.
-

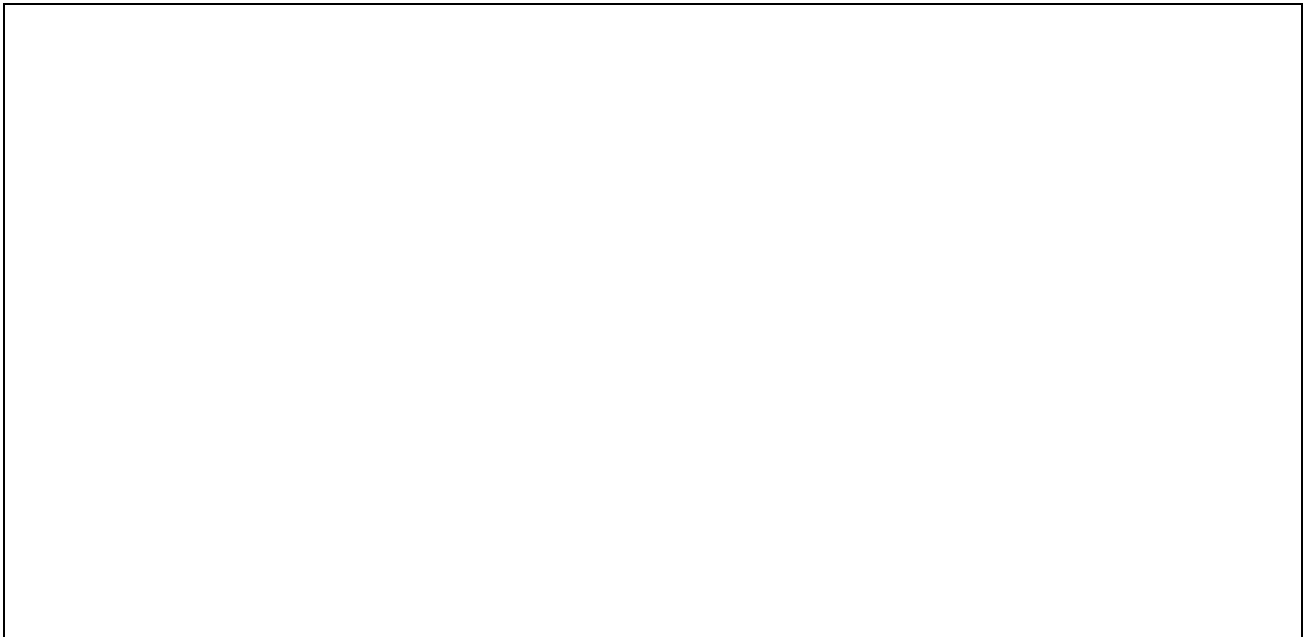
1. (15 points - **Correctness**) Huffman Encoding

(a) (5 points - **Correctness**) Create a Huffman Tree on the following distribution of characters:

{‘a’: 30, ‘c’: 40, ‘e’: 41, ‘h’: 82, ‘i’: 87, ‘n’: 52, ‘o’: 23, ‘r’: 68, ‘s’: 44, ‘t’: 75, ‘ ’:16}

(the last character is a space, you can free to call it ‘space’ or ‘ ’ in you tree).

At each split, clearly indicate which side gets ‘0’ and which gets ‘1’. At each split, you should assign ‘0’ to the subtree with the lower total weight and ‘1’ to the other subtree. If both subtrees have the same weight, you may assign ‘0’ and ‘1’ arbitrarily.



(b) (5 points - **Correctness**) Using **your** above tree, encode the sentence “this is a secret”.

Using **your** above tree, decode the sequence ‘00101111011101110001110111000011100010111100’. (If there are trailing bits at the end of your decoding, explicitly state what the trailing part of the sequence is.)



(c) (5 points - **Correctness**)

What is the runtime of building a huffman encoding?

Once you already have an encoding, what is the runtime of encoding a document?

Once you already have an encoding, what is the runtime of decoding a document?

Define the variables you think you need. If you feel you need (reasonable) clarifying assumptions, state them next to your answer. (Different answers will be allowed depending on your assumptions, but incorrect answers will not be given credit).

2. (15 points - **Correctness**) Suppose you've been given the following code:

```
class AVLNode {
public:
    AVLNode * left, right, parent;
    int value;
    int height; // root node stores tree height; a single node tree has height 1
};

//Rotate left @ top.
//E.g. top->parent should remain an ancestor of all of its descendants
//and top's location in the tree should be replaced by top->right
void rotateLeft(AVLNode * top);

void rotateRight(AVLNode * top) { /* Implementation not shown */ }
```

(a) (10 points - **Correctness**) Implement rotateLeft in either psuedocode or C++ (not English):

(b) (5 points - **Correctness**) Given the following method signature:

```
//Double rotate @ top
//so that top->parent remains an ancestor of all of its descendants
//and so that this method would be appropriate to call for
//balancing a tree consisting of three nodes:
// {top, top->left, top->left->right}
//(though this should work when there are more nodes in the tree)
void doubleRotate_LeftKink(AVLNode * top);

void doubleRotate_RightKink(AVLNode * top) { /* Implementation not shown */ }
```

Implement `doubleRotate_LeftKink` in either pseudocode or C++ (not English). You may assume the setup on the previous page, and that `rotateLeft` and `rotateRight` have been implemented:

3. (5 points - **Completeness**) Implement insert for an AVL tree in either pseudocode or C++ (not English). You may assume the code from problem (2).

```
//returns false if data is already in the tree, otherwise inserts  
//data into the tree and returns true.  
bool insert(AVLNode * root, int data);
```

4. (1 point - **Completeness**) Why, historically, are the colors in RB trees Red and Black?

5. (5 points - **Completeness**) Implement insert for an RB tree in either psuedocode or C++ (not English). You may assume the code below. Assume that the rotation methods correctly change all the pointers, but do not change the colors. You may use both this and the next page for your answer.

```
class BSTNode {
public:
    BSTNode * left, right, parent;
    int value;
    bool color; //false is red, true is black
};

void rotateLeft(BSTNode * top) { /* Implementation not shown */ }
void rotateRight(BSTNode * top) { /* Implementation not shown */ }
void doubleRotate_LeftKink(BSTNode * top) { /* Implementation not shown */ }
void doubleRotate_RightKink(BSTNode * top) { /* Implementation not shown */ }

bool insert(BSTNode * root, int data);
```



