Exam 1 – Sample Questions

October 26, 2015

Variables whose domain is not specified explicitly range over the reals. So for example, if a function is defined as $f(x) = x^2$, then $\operatorname{dom} f = \operatorname{cod} f = \mathbb{R}$.

Discrete math

Sets

True or false:

- If $x \in A$ then $x \in A \cup B$
- The empty set has no subsets.
- The empty set is a proper subset of every set.
- $5 \in \{2x + 1 : x \in \mathbb{N}\}$
- The power set of the set $\{1, 2, 3, 4, 5\}$ has 25 elements.

Relations

Which of these relations are reflexive? Symmetric? Anti-symmetric? Transitive?

- $\{(x, y) : x < 2y\}$
- $\{(x, y) : x, y \in \mathbb{N}, 2x \leq y\}$

- For $x, y \in \text{SetOfAllHumans}$, $x \, R \, y$ iff x and y have the same birthday.
- $\{(x, y) : x, y \in \mathbb{Z}, x^2 = y^2\}$

Functions

Which of these relations are functions?

- $\{(x, y) : x^2 = y\}$
- $\{(x, y) : x^2 = y^2\}$
- $\{(x, y) : x = y^2\}$

Which of these functions are injective? Surjective? Bijective?

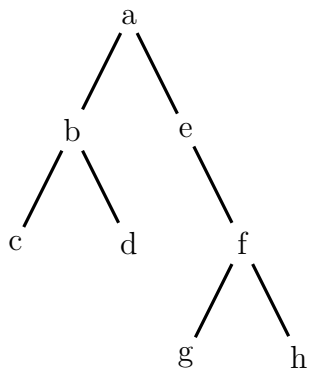
- $f(x) = x/2$
- $f : \mathbb{Z} \to \mathbb{Z}$ where $f(x) = 2x$.
- $f : [0, 1] \to [0, 1]$ where $f(x) = x^2$.

Huffman coding

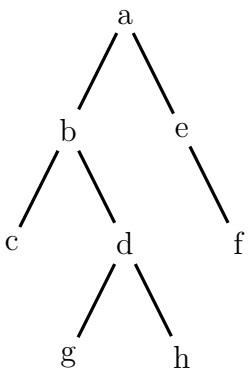
Trees

Consider these three binary trees:

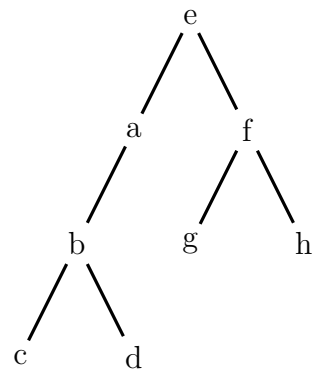
$T =$



$U =$



$V =$



Which pairs of them have the same:

- pre-order traversal?
- in-order traversal?
- post-order traversal?
- level traversal?

Write out the letters in the order they would be visited by each kind of traversal.

True or false:

- A binary tree node has exactly two children.
- A binary tree node has at most two children.
- A binary tree has edges labeled with 0s and 1s.

Prefix codes

Suppose we want to encode the symbols $\{A, B, C, D\}$ in binary. Is this a valid prefix code?

$$A \mapsto 0$$
$$B \mapsto 10$$
$$C \mapsto 110$$
$$D \mapsto 111$$

If it is, build the prefix code tree; if not, explain why you can't.

Encode this message: "BADCAB"

Decode this message: "11101110"

True or false:

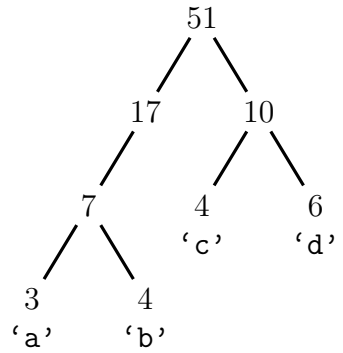
- A prefix code is a code in which every code word is a prefix of some other code word.

- Every code word in a prefix code must have the same length.
- If all the code words in a prefix code are reversed, the result is still a prefix code.

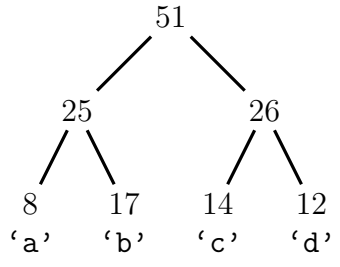
Huffman trees

Consider these two trees:

$H =$



$I =$



Which of these are valid Huffman trees? For those that aren't, what's the problem? Fix them.

Huffman code this message: "`brevity is the soul of wit`". Construct a Huffman tree based on the frequencies, and include the tree in your answer.

Arrays and linked lists

True or false:

- The time to get an element of an array, given its index, does not depend on the size of the array.
- One advantage of arrays over linked lists is that their capacity can grow in place without having to reallocate or copy elements.
- Unlike lists, the size of an array must be known at compile time.
- The time to get an element of a linked list, given its ordinal position in the list, does not depend on the position of the element.

- An array can use less space than a linked list because it does not need extra space for pointers.

A stack is a kind of data structure that allows adding and removing elements, and keeps track of the order that they are added so that they are removed in the reverse order. It has two main operations:

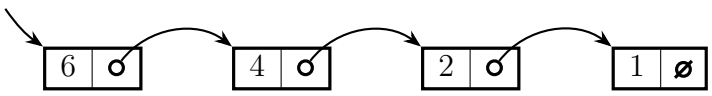
- The *push* operation adds an element to the queue.
- The *pop* operation the most recently *push*ed element that has not yet been popped.

For example, here is a sequence of interactions with a stack:

```
s = new Stack();        // s is empty

s.push(1);              // s is 1
s.push(2);              // s is 2 1
s.push(3);              // s is 3 2 1

a = s.pop();            // s is 2 1, a is 3

s.push(4);              // s is 4 2 1
s.push(5);              // s is 5 4 2 1

b = s.pop();            // s is 4 2 1, b is 5

s.push(6);              // s is 6 4 2 1
```

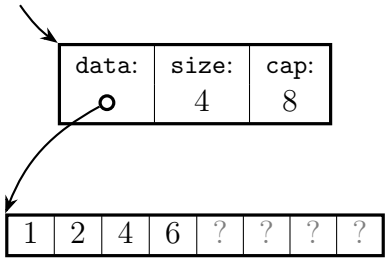
Now consider these two representations of a stack:

Linked list representation. The elements are stored in newest-to-oldest order in a singly-linked list. For example, here is how the stack `s` from the sample interaction would be represented:



The next element to be popped would be 6, followed by 4, 2, and then 1.

Array representation. The elements are stored in oldest-to-newest order in an array, along with two additional pieces of information: the current number of elements in the stack (labeled `size`) and the capacity of the array (labeled `cap`). For example, here is stack `s` in the array representation:



As in the other representation, the next element to be popped would be 6, followed by 4, 2, and then 1. The light gray question marks (?) indicate locations in the array whose values don't matter, either because those values have been popped or because those locations haven't been used yet.

Questions:

- What are some advantages and disadvantages of each representation?
- In your preferred language (or clear pseudocode), implement these operations:
  - *push* for the linked list representation
  - *pop* for the array representation

  If the stack is empty then *pop* should signal an error. (If you don't know how, writing a comment that says "`signal empty stack error here`" is sufficient.) You may assume that each structure or class has a constructor that takes all the fields and stores them in the new object. Clearly state any other assumptions that you make.