Contents

1 What I did today

1. Use Haskell package to simplify Html multiple lines in Haskell *<2019-05-09 Thu>*

   ```
   package Text.RawString.QQ

   multipline = [r|
        <p>"dog"</p>
        <p>"cat"</p>
        |]
   ```

2. Added code to process background process in Shell from Web Input in wai server *<2019-05-10 Fri>*

   ```
   "x nohup filewatcher &"
   ```

3. ghci does not load package into Ghci if you use :load myhaskell.hs It causes lots of confusing when you load your code inside ghci

2 Added Javascript to edit code block

1. Added Javascript to listen to onclick event, e.g. enable/disable TextArea visibility
2. Hide header to TextArea with css `display:none`
3. Generate id for each code block in TextArea
4. Go through each code block and remove the edited code block from a list
5. Append the edited the code block to the new list

3 MacOS automatically update screws up tmux

1. **tmux** can not start up

   - reinstall tmux:

     ```
     brew uninstall --force tmux
     brew install tmux
     ```

‘’ ‘ ‘dd vim script to check definition of Haskell function with Timer and `job_start()`

1. Create a timer which calls a function every n seconds, e.g. 0.5 second `timer_start()`
2. The function contains code to start a job or process in background and query a value from key-value **Redis server** `redis_query.hs` The key is the current **word** under current **cursor**
3. The function calls `systemlist()` and passes `$b/redis_query.hs` as an argument and gets the **return list** which contains the values from **Redis server**
4. Use `echo` to display the **return list** at the bottom of **Vim**

4 Haskell show, print and putStr or putStrLn

1. show, print and putStr `"dog\\"`,

   ```
   show "dog\\" -- "\"dog\\\\\""
   print "dog\\" -- "dog\\"
   putStr "dog\\" -- dog\
   ```

5 Add wai-uil to wai.hs Server for redirecting response URI

1. Install wai-uti with `cabal install wai-uti`
2. Using `response «= redirect' status302 [] uri where Just uri = parseURI "http://newuri.com"` wai.hs

6 Solve "Insert a node into a binary tree", it seems to be an easy question, but it is tricky to get it right on the first try.

- Two ways to solve the problem.
  1. Use **Iteration** with `while`
     - Start from the root node as current node. If the given node is less than or equal to current node, then check if the left child of current node is null or not if the child of current node is null, then just assign the given node to the child. Done else using `tmp = tmp.left` to goto next child of current node. otherwise repeat the same thing in the right subtree.
  2. Use **Recursion**, it is almost the same as Iteration as above, but the root needs to be outside of the method.
- Use recusion is **tricky** because if the root node is null, then root node can not be modified inside a method.
- The best way to do is to use **Iteration**

```
   // Iteration
public static void insert(Node node){
    Node curr = root;
    // left subtree
    if(node.data <= curr.data){
if(curr.left == null)
    curr.left = node;
else
    curr = curr.left;
    }else{
// right subtree
if(curr.right == null)
    curr.right = node;
else
    curr = curr.right;
    }
}

   // Recursion
```

```
public static void insert(Node root, Node node){
    if(root != null){
if(node.data <= root.data){
    if(root.left == null)
root.left = node;
    else
insert(root.left, node);
}else{
    if(root.right == null)
root.right = node;
    else
insert(root.right, node);
}
    }
}
```

6.1 DONE hair cut

1. 17$ for the stupid hair cut.

6.2 DONE fix add, update and delete buttons on wai.hs

1. There is some issue in with runSh "mv" to rename file, after remove "mv" it seems to be working

6.3 DONE Do my laundry

6.4 DONE Install Vim 8.1.135, and fix symbol link with the Vim syntax.vim file error

6.5 DONE Try to understand Visitor pattern in Java and Haskell Algebraic Datatypes

6.6 DONE Add Redirect response to wai.hs with wai-uti

6.7 DONE pushall respositories to bitbucket

6.8 DONE fix resume a bit

6.9 TODO Need to add more detail to resume

6.10 TODO Buy a new pot and buy some rice

6.11 DONE Open new paypal account

6.12 DONE Add color to wai.hs to highlight the current edited code block

1. when current code block is clicked, change the background of pre
2. Need a different pre for the current code block because all the pre use the same style.
3. Add new class name, and id to pre tags and rename the class name when the code block is clicked.

6.13 DONE Send out more resumes

6.14 TODO Do some java coding

- Rotate square array to left/right 90 degrees.
- Balance brackets.
- Rotate spiral rectangle.
- Insert a node into a binary tree.
- Find all the nodes that are distance k from a given node
- Check whether a node is in the binary tree or not

- Single linked, append, addFront and remove operations, it seems to be working
- Added html file filter to genehtml.hs to avoid error when genehtml reads non-html file.
- Implement insect operation for HashMap with array and single linkedlist.
    1. Remember to increase the count if a node is inserted
    2. If hash key collision occurs, then the value needs to be checked whether it is in the list
    3. When walking through the list, previous node need to be kept.
    4. Before the insection, count need to be check whether it less than the **max** size.
- Implement **HashMap** insect operation using binary tree.
- Check a binary tree is whether a **BST**.
    1. Use backtrack.
    2. Use **BST** definition.
- Added PriorityQueue code example, `java.util.PriorityQueue`, and use lambda for comparatoin,. Add code using `Comparable<Person>` and `Comparator<Person>`

```
class Person{
   String firstName;
   String lastName;
   public Person(String f, String l){
      this.firstName = f;
      this.lastName = l;
   }
}

// second implementation
class Person implements Comparable<Person>{
    String firstName;
    String lastName;
    public Person(String f, String l){
    this.firstName = f;
```

```
    this.lastName = l;
   }
   public int compareTo(Person other){
     return this.firstName.compareTo(other.firstName);
   }
}

PriorityQueue queue = new PriorityQueue((\a, b) -> a.firstName.compareTo(b.firstNa
queue.add(new Person("David", "lee"));
queue.add(new Person("Tommy", "kee"));
queue.add(new Person("Jacky", "kuu"));
while(!queue.isEmpty()){
   Person p = queue.remove();
   System.out.print(p.firstName);
}

// use Comparator interface,
class PersonCmp implements Comparator<Person>{
     public int compare(Person p1, Person p2){
       return p1.firstName.compareTo(p2.firstName);
     }
}
Collections.sort(new PersonCmp());
```

- Added merge two sorted lists.
- Added Lease Recent Used code.
- Added Double linked List, delete, append and insertFront.
- Read file line by line and split them.
- Serialize and deserialize binary tree with map.
  - Use map with $k = 0, 2 * k + 1, 2 * k + 2, \ldots$ to index binary node.
  - File, BufferedReader, BufferedWriter, `BufferedReader br = new BufferedReader(new FileReader(fname))`
  - Use **preorder** traveral to write all the keys and nodes to file.

    ```
    public void serialize(Node curr, Integer k, BufferedWriter bw){
        // preorder traveral
        // handle IOException here
    ```

```
    // k=0, k=1, k=2, k=3 ...
    bw.write(k + " " + curr.data + "\n");
    serialize(curr.left, 2*k + 1);
    serialize(curr.right, 2*k + 2);
}

public Node deserialize(Integer k, Map<Integer, Integer> map){
    // handle IOException here
    Integer v = map.get(k);
    if(v != null){
Node root = new Node(v);
root.left = deserialize(2*k + 1, map);
root.right = deserialize(2*k + 2, map);
return root;
    }
    return null;
}
// other implementation, use iteration for preorder traveral to read node fro
```

- Use **preorder** traveral from map and build a tree **from bottom and up**.

6.15 TODO Take a look at the Algebraic Data Type and Visitor pattern in Java

URL

6.16 DONE Send out more resume, more to EA

6.17 DONE Add Applescript to send notification when filewatcher detects some txt file inside password folder.

6.18 DONE Finally figure out how to use multiple putStrLn in Agda

6.19 DONE Refactor filewatcher.hs, remove IORef, clean up useless code

6.20 DONE Added double click: `ondblclick` event to javascript file inside wai.hs to improve user experience.

6.21 DONE Finished osense take home assignment

1. Use C standard library to display statistic temperature info from a CVS file.

    - Read the CVS file line by line
    - Allocate memory to store all the temperatures in an array.
    - Write a quick sort function to sort the array
    - From the sorted array, mean, median, mode, min, max and standard deviation can be compuated.

6.22 DONE Added gcc compile script to vimrc file, use `$scr/gcc_compile.sh`

6.23 I work on Application generate report for business department

- The Applicaiton take data from database such as sale information,
- and we do accounting calculation based on the data.
- We generate report at the end of each months
- The report store in S3 which Amazon file storage,
- The business department can download the report from S3.
- Technologies that we used:
- The application is Java based.
- We use:

- Hiberate use database modeling
- Spring an Application framework, dependency injection, annotation, MVC Model View Controller

6.24 In VoiceBox Inc.

- Development Application to generate nature language.
- e.g. Given a set of rules, The App generate English language based on them.
- e.g. What is the temperature in Vancouver Today,
- Vancouver is a city, and
- The set of rules is like "What is the temperature in City today"
- What is the intentation of user,
- e.g. do I need a jacket today?