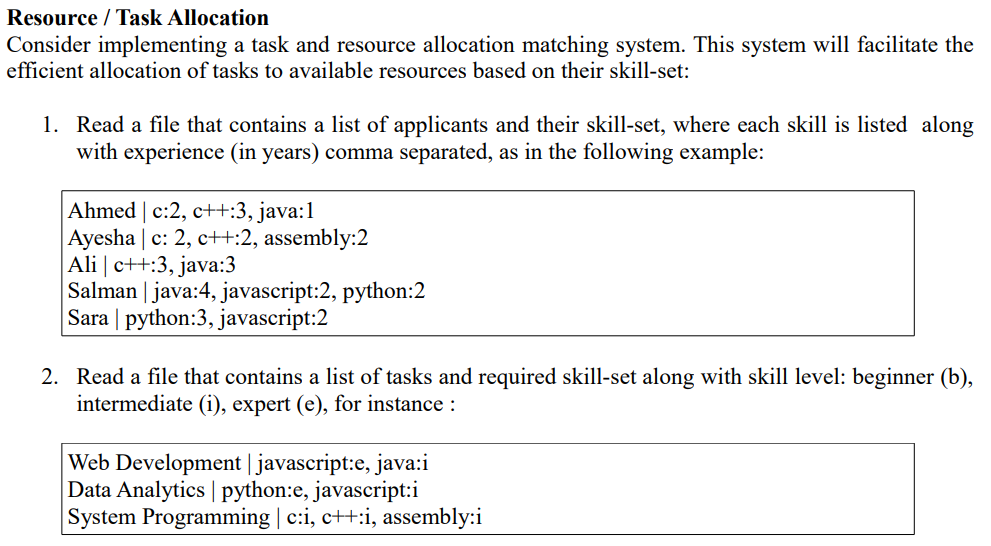
**Data Structures**

**Assignment 3**

Question no. 1

We have implemented a linked list of linked lists, stack of queues. Now it’s time to implement tree of trees where each node of tree will have root node of another tree. Let’s understand with an example.

**Example:**



Here b = 1, i = 2, e = 3. We are having 3 types of data- skill, resource and task mean we will have 3 types of trees skillTree, resourceTree and taskTree. Their structure **must be** like as follows:

class skillTree {

class skillNode {

int level; // it will be used as integer value in node to sort skill tree

string name;

}

skillNode \*left;

skillNode \*right;

}

class resourceTree {

class resourceNode {

int id; // it will be used as integer value in node to sort resource tree

string name;

skillTree \*root;

}

}

class taskTree {

class taskNode {

int id; // it will be used as integer value in node to sort task tree

string name;

skillTree \*root;

}

}

Let’s look at an example in which have the following data set.

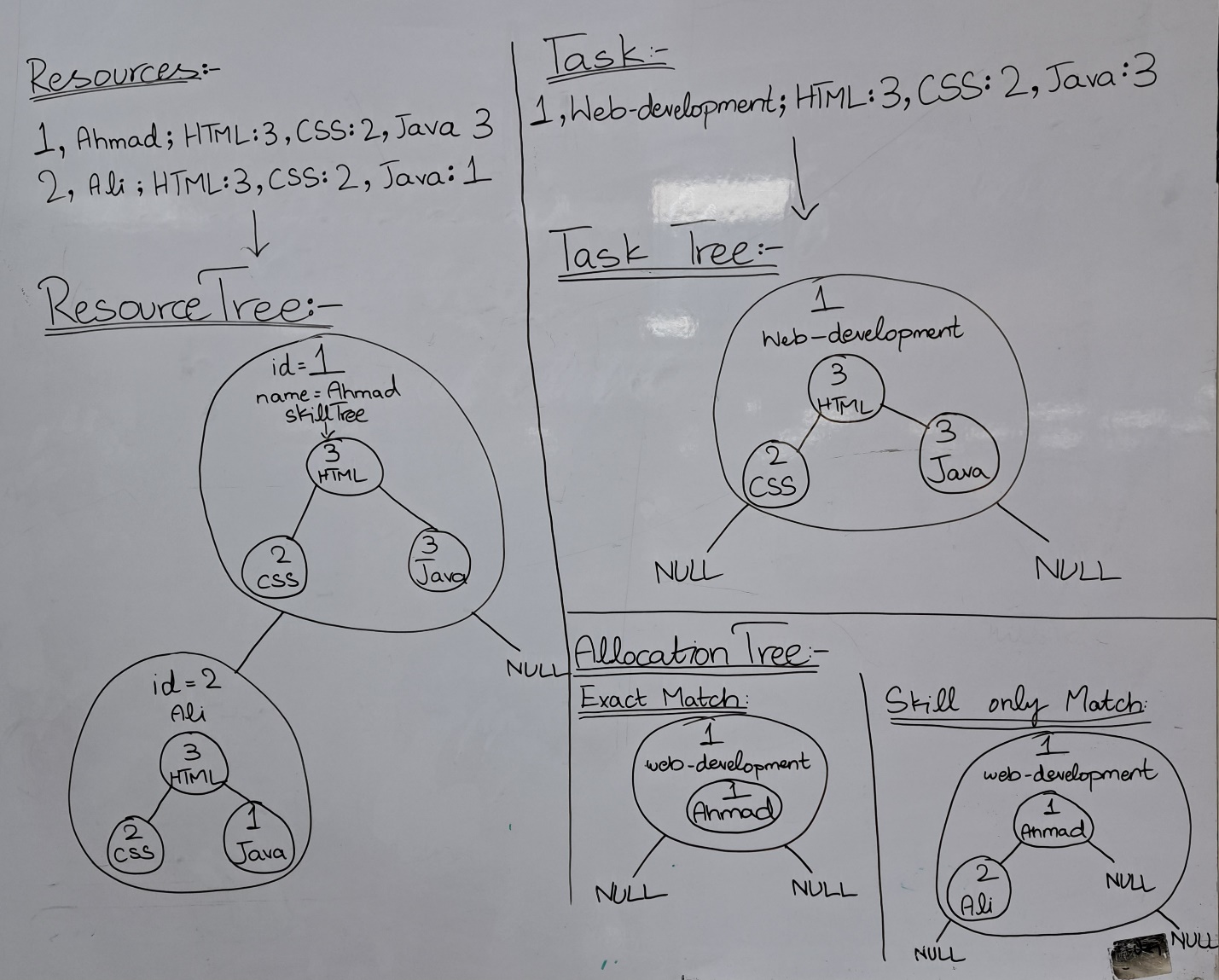
task tree is having 1 node {id = 1, name = web-development, skillTree = {{name = HTML, level = 3}, {name = CSS, level = 2}, {name = Java, level = 1}}}

resource tree is also having 1 node {id = 1, name = Ahmad, skillTree = {{name = HTML, level = 3}, {name = CSS, level = 2}, {name = Java, level = 3}}}

you will start by traversing task tree, at each node there is task name to which you must allocate tasks by matching skill tree. At each node there is a skill tree which has skill name and level, if the level here is 3 then you will sort skill tree at each node of resource tree with max heap so that you will have level 3 skills at top of skill tree in resource tree, if you found level 2 skill to match, do not sort but in case you find 1, you will sort skill tree at each node in task tree by using min heap so that skills with level 1 will be at top of tree.

After applying heap sort algorithm, you will match skill level with skill level at first node of skill tree in first node of resource tree, if it matches then you will match name other wise you will go to next node of skill tree in first node of resource tree. You will do the same for all nodes of resource tree with one node of task tree and make a tree of allocations which will have the allocated resources for each task.

**Example visualization:**



You will match and make the allocation tree as above.

class AllocationTree {

class AllocationNode {

int id;

string name;

class allocatedResources {

int id;

string name;

}

}

}

At the end you must encode the allocation tree with the help of Huffman encoding. You will get the file size the same even after the encoding if you do it with strings or characters. You need to convert the allocation tree in binary tree values then you must apply encoding and decoding algorithms on allocation tree with binary values. That’s it.

**Note:**

1. You must traverse skill trees by implementing level order traversal.
2. Skill tree in resource tree must by traversed by first sorting with min or max heap according to requirements mentioned above each time.
3. You can convert your nodes of allocation tree into tree of binary values at each node by converting each character in name to 8-digit binary value and concatenating them.
4. You will encode the tree of binary values by using Huffman encoding and put both values before encoding and after encoding in text file and your size of encoded text file must be less than decoded file.
5. You must implement two strategies **exact match** (will compare exactly name and level of skills) and **skill only match** (will only match names of skills)