

Haskell Coursework Report

Q13

- Screenshot:

```
*Main> search1 = findTransFrom 10 f1 [Pear, Plum, Pear, Apple]
*Main> search1
(-1, 2, [(1, 0), [Pear, Apple, Plum], (0, (Rm, 0), [Apple, Plum]), (0, (Rm, 1), [Pear, Plum]), (0, (Rm, 2), [Pear, Apple]), (0, (Dup, 0), [Pear, Pear, Apple, Plum]), (0, (Dup, 1), [Pear, Apple, Apple, Plum]), (0, (Dup, 2), [Pear, Apple, Plum, Plum]), (0, (Exc, 0), [Apple, Pear, Plum]), (0, (Exc, 1), [Pear, Plum, Apple]), (1, (Rm, 0), [Plum]), (1, (Rm, 1), [Apple]), (1, (Dup, 0), [Apple, Apple, Plum]), (1, (Dup, 1), [Apple, Plum, Plum]), (1, (Exc, 0), [Plum, Apple])])
*Main> transPath search1
[]
*Main>
```

- Answer:

`search1 = findTransFrom maxiter f1 f2`

`where maxiter = 10, f1 = [Pear, Apple, Plum], f2 = [Pear, Plum, Pear, Apple].`

- For question: What does the output of this function call mean?

The output for `search1` is a `(f2_pos, (pos', tree'))` pair in which `f2_pos` is the position index of node in current `tree'` which outcome (fruit list) is equal to the target fruit list `f2`. `(pos', tree')` is a pair where `pos'` is position of node to be expanded next, `tree'` is a node list that stores all 3-tuples nodes that is in the tree.

In this question, output for `f2_pos` is `-1`, which means that when the size of tree is **right** larger than `maxiter` (10), there is no node with outcome that is equal to target fruit list `f2` (`[Pear, Plum, Pear, Apple]`). Output for `pos'` is 2 which means the next node to be expanded is the node with position index 2 in node list (`tree'`). That is, nodes in node list `tree'` with index 0 and 1 has already been expanded before size of tree exceeds `maxiter` (10). Output for `tree'` contains a list of nodes in the format of 3-tuples, including all the nodes that has been expanded before the expansion of tree stops (parent nodes) and all their child nodes in BFS order.

- For question: Has it found the solution? If so, what is it? If not, why not?

It has **NOT** found the solution. The first reason is that, as I mentioned above, output `f2_pos` is `-1`, which means that no node with an outcome that is equal to target fruit list `f2` (`[Pear, Plum, Pear, Apple]`) is found in current tree. Therefore, no position index of node can be returned, but only `-1`. Another reason is that, as the value of `maxiter` is set quite small, the expansion of tree

stops before a node with an outcome that is equal to f2 is discovered.

Q16

- Screenshot:

```
*Main> search2 = findTransFrom 10000 [Apple, Pear, Banana, Orange] [Apple, Pear, Apple, Banana]
*Main> transPath search2
[((Id, 0), [Apple, Pear, Banana, Orange]), ((Rm, 3), [Apple, Pear, Banana]), ((Dup, 0), [Apple, Apple, Pear, Banana]), ((Exc, 1), [Apple, Pear, Apple, Banana])]
*Main> _
```

```
*Main> search2 = findTransFrom 10000 [Apple, Pear, Banana, Orange] [Apple,
Pear, Apple, Banana]
```

```
*Main> transPath search2
```

```
[((Id, 0), [Apple, Pear, Banana, Orange]), ((Rm, 3), [Apple, Pear, Banana]), ((Dup, 0),
[Apple, Apple, Pear, Banana]), ((Exc, 1), [Apple, Pear, Apple, Banana])]
```

- Answer:

For question: is it the same solution as given in Table 1?

My solution is the **same** as given in Table 1. My solution has 3 steps as shown above:

Step	Operation	Row
--	Starting	Apple,Pear,Banana,Orange
1	Remove 4 th fruit	Apple,Pear,Banana
2	Duplicate 1 st fruit	Apple,Apple,Pear,Banana
3	Exchange 2 nd fruit with its right neighbor	Apple,Pear,Apple,Banana

The initial fruit list is set to [Apple, Pear, Banana, Orange], which is the same as Table 1, as well as the target fruit list [Apple, Pear, Apple, Banana]. Also, I assign 10000 to maxiter, which is large enough to successfully find a node with outcome that is the same as target fruit list. Then, I make use of function “transPath” to show the path of transactions and fruit list at each stage of the solution, which is exactly same as table 1.