

Haskell Assignment 1

Test Results

listToBag

`listToBag [] → []`

`ListToBag[1] → [(1,1)]`

`listToBag[1,1,1] → [(1,3)]`

`listToBag[1,40,27,5] → [(1,1),(40,1),(27,1),(5,1)]`

`listToBag [1, 1, 2, 35, 2, 4, 85] → [(1,2),(2,2),(35,1),(4,1),(85,1)]`

bagEqual

`bagEqual (listToBag []) (listToBag []) → True`

`bagEqual (listToBag [1])(listToBag []) → False`

`bagEqual (listToBag [34]) (listToBag [34]) → True`

`bagEqual (listToBag [2]) (listToBag [5]) → False`

`bagEqual (listToBag ["x", "y"]) (listToBag ["x", "y"]) → True`

`bagEqual (listToBag ["x", "y"]) (listToBag ["x", "z"]) → False`

`bagEqual (listToBag [1, 2, 3]) (listToBag [1,2,3]) → True`

`bagEqual (listToBag [1, 2, 3]) (listToBag [4, 5, 6]) → False`

`bagEqual (listToBag ["foo", "bar"]) (listToBag ["bar", "foo"]) → True`

`bagEqual (listToBag ["foo", "x", "bar"]) (listToBag ["bar", "foo"]) → False`

bagInsert

`bagInsert [] (listToBag []) → [([],1)]`

`bagInsert 0 (listToBag []) → [(0,1)]`

`bagInsert 1 (listToBag [1,2]) → [(1,2),(2,1)]`

`bagInsert 3 (listToBag [1,2]) → [(1,1),(2,1),(3,1)]`

bagSum

`bagSum (listToBag []) (listToBag []) → []`

`bagSum (listToBag [1]) (listToBag []) → [(1,1)]`

`bagSum (listToBag [1]) (listToBag [1]) → [(1,2)]`

`bagSum (listToBag [1,2,25,3,2]) (listToBag [25,1,2,4]) → [(4,1),(3,1),(2,3),(25,2),(1,2)]`

bagIntersection

`bagIntersection (listToBag []) (listToBag []) → []`

`bagIntersection (listToBag [1]) (listToBag []) → []`

`bagIntersection (listToBag [1,2,4]) (listToBag [1,2,3]) → [(1,1),(2,1)]`

`bagIntersection (listToBag [1,4,2,4]) (listToBag [1,2,3,4,4]) → [(1,1),(4,2),(2,1)]`

itemCount (helper function)

`itemCount [] (listToBag []) → 0`

`itemCount 1 (listToBag []) → 0`

`itemCount 1 (listToBag [1]) → 1`

`itemCount 4 (listToBag [4, 2, 3, 4]) → 2`

bagInsertN (helper function)

`bagInsertN 1 0 (listToBag []) → []`

`bagInsertN 1 1 (listToBag []) → [(1,1)]`

`bagInsertN 1 2 (listToBag []) → [(1,2)]`

`bagInsertN 1 2 (listToBag [4,3,2]) → [(4,1),(3,1),(2,1),(1,2)]`