Used real data from Kaggle: <https://www.kaggle.com/jwaitze/tablesoftheelements/data>

Table contains information on all elements in periodic table. Has 117 tuples. Includes such information as density, mass, allotropes and etc.

I use B-tree of order 4. I am using name of element(column B) and it’s atomic mass(column K) as key values. Name has more priority.

Programm can be tested as followings:

1. Firstly, an instance of tree should be created:

tree = Btree()

1. We can test INSERT by calling push() function on tree. In the slides tuples where inserted by number(value). So, in column A of xls file there is enumeration for each element. They represent value for each search key. For example, Hydrogen has value 1. Push() accepts integer from 1 till 117. It is called like this: tree.push(i) . i is an integer
2. We can test DELETE by calling delete() function on tree. In the slides tuples where deleted by number(value). So, in column A of xls file there is enumeration for each element. For example, Hydrogen has value 1. Delete() accepts integer from 1 till 117. It is called like this: tree.delete(i) . i is an integer
3. We can test SEARCH by calling search() function on tree. In the slides tuples are searched by searching key in tuple form. So, search() accepts tuple. For example, (“Hydrogen”, 1.008) is an example of searching key. It is called like this, for example: tree.search(("Boron", 10.81)) , which will return 5 as it’s its value. In the slides values are returned in the list form, so return value will be [5] to be precise.
4. We can test UPDATE by calling update() function on tree. It accepts value of searching key to be changed and tuple of values which will become new searching keys. Update() function causes changes not only in the tree, but also in the periodic\_table.xls file (xlrd, xlwt, xlutils were imported for making this.) Example: tree.update(1, ("Hyd", 1111))

Example code: tree = Btree()

for i in range(1,100):

tree.push(i)

print(tree.search(("Boron", 10.81)))

for i in range(1, 55):

tree.delete(i)

tree.update(1, ("Hydro", 1.008))

tree.printtree() #prints 1st two levels of tree