FACULTY OF ENGINEERING DESIGN AND TECHNOLOGY

NAME: MOGA MUZAMIL ABDUL WAHAB

REG NO: S21B23/013

ACCESS NO: A94166

COURSE: BACHELOR OF SCIENCE IN COMPUTER SCIENCE (BSCS)

COURSE UNIT: DESIGN AND ANALYSIS OF ALGORITHMS

LECTURER: MR. JOSEPH NDAEBWA WAMBETE

```python
def search(list,values):

    for i in list:

        for j in i:

            if values == j:

                return i

    return "Not in list"


def main():

    list = [["A96447","MUGANGA Charles","J22B23/032"],["A95681","NAJJOBA
Tracy","S21B23/034"],["A94169","KATUKUNDA Rochelle","S21B23/016"],["A94160","MUKISA
Isaiah","S21B23/007"],["A94161","NKATA Joshua Luyombya","S21B23/008"],["Afghanistan", "93"],
["Fiji", "679"], ["Bahamas", "1-242"],["Tanzania", "255"], ["Saint Vincent and the Grenadines",
"1784"],["Ukraine", "380"]]


    countryCode = "7"

    newList = search(list,countryCode)

    print("I am searching for",countryCode,"and is found",newList[0], "is the country")


    accessNumber="A94160"

    newList = search(list,accessNumber)

    print("I am sarching for",accessNumber, "and is found",newList[1], "is the name",newList[2], "is the
registration number.")


    countryCode = "380"

    newList = search(list,countryCode)

    print("I am searching for",countryCode,"and is found",newList[0], "is the country")


    name="Doe"

    newList = search(list,name)

    print("I am sarching for",name, "and is found",newList[0], "is the Access Name",newList[2], "is the
registration number.")
```

name="KATUKUNDA Rochelle"

newList = search(list,name)

print("I am sarching for",name, "and is found",newList[0], "is the Access Name",newList[2], "is the registration number.")

Country="Tanzania"

newList = search(list,Country)

print("I am sarching for",Country, "and is found",newList[1], "is the countryCode")

main()

COMPLEXITY OF THE SEARCH ALGORITHM

Using the Aggregate method to calculate the big O of the algorithm.

def searching(list, values): O(1)

    for i in list: O(n)

        for j in i:#i being the nested list O(n)

           if values == j: O(1)

              return i O(1)

         return "Not in list" O(1)

        $O((n+1)*(n+1+1+1) = O(n^2+3n+n+3)$

        $=O(n^2 + 4n + 3)$

therefore the Big O is O(n) is the worst case since no swapping is done but a program can terminate at any point when the checking through the list.