

Report

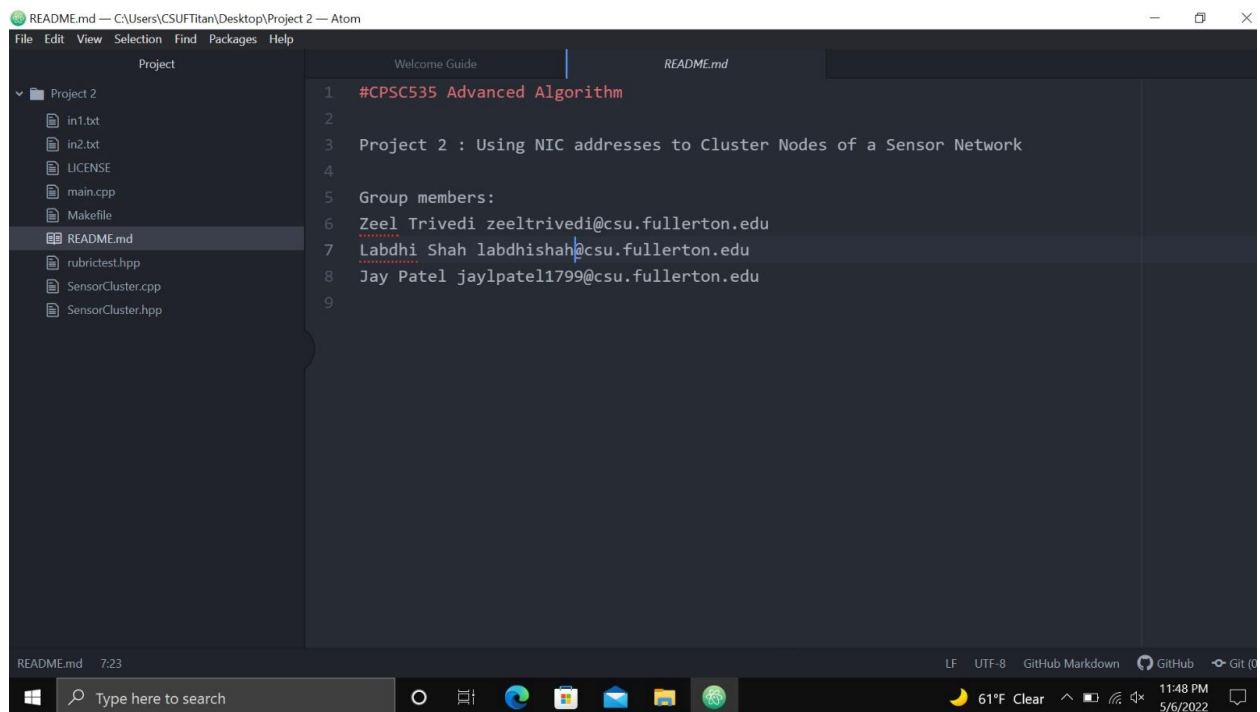
Project 2

Group Members

Zeel Trivedi : zeeltrivedi@csu.fullerton.edu

Labdhi Shah : labdhishah@csu.fullerton.edu

Jay Patel : jaylpatel1799@csu.fullerton.edu



```
1 #CPSC535 Advanced Algorithm
2
3 Project 2 : Using NIC addresses to Cluster Nodes of a Sensor Network
4
5 Group members:
6 Zeel Trivedi zeeltrivedi@csu.fullerton.edu
7 Labdhi Shah labdhishah@csu.fullerton.edu
8 Jay Patel jaylpatel1799@csu.fullerton.edu
9
```

Pseudocode description:

There are total 6 hash tables in given program.

Pseudocode for the given problems:

##the given has value is of six digit

##so to compute it the first digit of the address, the module logic is used

to compute first digit, $(\text{address1}/100000 \% 10)$

to compute second digit, $(\text{address2}/10000\%10)$

to compute third digit, $(\text{address3}/1000\%10)$

to compute fourth digit, $(\text{address4}/100\%10)$

to compute fifth digit, $(\text{address5}/10\%10)$

to compute sixth digit, $(\text{address6}/1\%10)$

##now after returning the hash value based on the each digit,

##copying the text, from one file to other file.

##now function to add the NIC to the given sensor network file for example, using the hash table,

hashTable1.insert({nic1, item1})

hashTable2.insert({nic2, item2})

hashTable3.insert({nic3, item3})

hashTable4.insert({nic4, item4})

hashTable5.insert({nic5, item5})

hashTable6.insert({nic6, item6})

#now using delete function to remove the NIC value from network, if the function finds the NIC in file, then it deletes the value; else it will return the false and close the file.

hashTable1.erase(nic)

hashTable2.erase(nic)

hashTable3.erase(nic)

hashTable4.erase(nic)

hashTable5.erase(nic)

hashTable6.erase(nic)

##function will decide the best hash function from the six hasing table and the most balanced sensor network for the current set of NIC address to find the maximum and minimum value from each hashtable turn by turn

```
unsigned int minimum_val = hashTable1.bucket_size(0);
```

```
unsigned int maximum_val = hashTable1.bucket_size(0);
```

```
for (auto i = 1; i < 10; ++i){
```

```
    unsigned int bucket = hashTable1.bucket_size(i);
```

```
    if (bucket < minimum_val){
```

```
        min = bucket;
```

```
    } else if (bucket > maximum_val){
```

```
        max = bucket;
```

```
##now to find the best, best_value = maximum_value - minimum_value
```

Result:

```
hash function 1 on item 123456 returns 1::1
hash function 2 on item 123456 returns 1::2
hash function 3 on item 123456 returns 1::3
hash function 4 on item 123456 returns 1::4
hash function 5 on item 123456 returns 1::5
hash function 6 on item 123456 returns 1::6
New network. Size is 2 after adding two NICs: Velocity sensor 123456 and Particle sensor 234567
opening file in1
BestHashing() for in1.txt returns:: 2
opening file in2.txt
BestHashing() for in2.txt returns:: 2
New network then read in2.txt. Then remove two NICs: 110987 and 210FED. Size becomes 35
BestHashing() for in2.txt returns:: 3

...Program finished with exit code 0
Press ENTER to exit console.[]
```