Hashing

Anh Huynh

1. Assume you have a simple single-dimensional array

```
array = [200, 400, 100, 50, 350]
```

Linear search will take O(N). Write a code in C++/Python to improve the search operation efficiency from O(N) to O(1). **4 pts**

```cpp
#include <iostream>
#include <unordered_set>
using namespace std;


int main(){
    // [200, 400, 100, 50, 350]
    // create a hash table named list
    unordered_map<int, string> list;

    // store key-value pairs in unordered
    list[200] = "one";
    list[400] = "two";
    list[100] = "three";
    list[50] = "four";
    list[350] = "five";

    // search for this key
    int key = 100;

    // if the key exists in the hash table, then print its value
    if (list.find(key) != list.end()){
        cout << "Key found: " << list[key] << endl;
    }
    // key doesn't exist
    else {
        cout << "Key not found" << endl;
    }

    return 0;

}
```

2. Use C++, generate hash value of your name. **1 pts**

3. With the help of a figure, explain the problem that occured due to introducing a **tombstone** to mark the deleted cell. **5 pts**