Homework week 5

Set, Hash and Map

Câu 1: What are two desirable properties of a hash function?

* Xác định: Một hàm băm tốt nên luôn tạo ra cùng một giá trị băm cho cùng một đầu vào. Điều này có nghĩa là nếu bạn băm cùng 1 dữ liệu nhiều lần, bạn sẽ luôn nhận được mã băm giống nhau mỗi lần. Sự xác định là quan trọng để đảm bảo tính nhất quán và dự đoán trong các ứng dụng khác nhau, chẳng hạn như truy xuất dữ liệu và xác minh tính toàn vẹn của dữ liệu.
* Phân phối đều:Một hàm băm tốt nên phân phối giá trị băm đều trong không gian đầu ra của nó. Điều này có nghĩa là các giá trị băm nên phân bố đều trên khoảng các mã băm có thể có. Sự phân phối đều này giúp ngăn chặn xung đột( tình huống hai đầu vào khác nhau tạo ra cùng một giá trị băm) và đảm bảo rằng hàm băm hiệu quả . Phân phối không đều có thể dẫn đến hiệu suất kém và làm suy yếu tính hiệu quả của hàm băm.

Câu 2: Draw the 11-entry hash that results from using the hash function h(i) = (2i+5) mod

11 to hash keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5.

a) Assume collisions are handled by chaining.

b) Assume collisions are handled by linear probing.

a,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 44 |  |  | 16 | 88 | 94 | 12 |  | 13 |  |
|  | 20 |  |  | 5 | 11 | 39 | 23 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

b,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 39 | 44 | 20 | 5 | 16 | 88 | 94 | 12 | 23 | 13 | 11 |

Câu 3:

Draw the 17-entry hash that results from using the hash function h(i) = (i+3) mod 17 to hash keys 1, 3, 18, 8, 23, 35, 11, 36, 20, 16. c) Assume collisions are handled by chaining. d) Assume collisions are handled by linear probing.

a,

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|  |  | 16 |  | 1 | 36 | 3 |  |  | 23 |  | 8 |  |  | 11 |  |  |
|  |  |  |  | 18 |  | 20 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 35 |  |  |  |  |  |  |  |  |  |  |  |  |

b,

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|  | 16 |  |  | 1 | 18 | 3 | 35 | 36 | 23 | 20 | 8 |  |  | 11 |  |  |

Câu 4:

#include<bits/stdc++.h>

using namespace std;

typedef struct Node\* node;

class Student {

private :

string Class;

string name;

public :

Student() {

Class = "NA";

name = "NA";

}

void setName(string \_name) {

this->name = \_name;

}

string getName() {

return this->name;

}

void setClass(string \_class) {

this->Class = \_class;

}

string getClass() {

return this->Class;

}

};

class Node {

int data;

Node\* next;

};

map<int, Student> mp;

void Insert(int ID, string Name, string Class) {

Student x;

x.setName(Name);

x.setClass(Class);

mp[ID] = x;

}

void INFO(int ID) {

for(auto tmp : mp) {

if(tmp.first == ID) {

cout << tmp.first << " " << tmp.second.getName() << " " << tmp.second.getClass() << endl;

return;

}

}

cout <<" NA" << ",NA" << endl;

}

void DELETE(int tmp) {

mp.erase(tmp);

}

int main()

{

Insert(1, "Tuan", "K61KS");

Insert(2, "Vuong Thi Hai Yen", "k55");

INFO(3);

INFO(2);

DELETE(2);

INFO(2);

return 0;

}