NCKU Programming Contest Training Course
STL (vector, map)
2016/03/02

Jian-Min Lin (riljian)

f74016043@mail.ncku.edu.tw

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Outline

vector

map

Hash

made by electron & free999 & riljian

vector

- Array
  - Unable to dynamically modify the data
- vector
  - Dynamic array
  - Insert/Erase an element at the specific index
- When
  - Sparse matrix
  - Adjacency list
  - …

made by electron & free999 & riljian

vector

- Useful member functions
  - operator[]
  - push_back()
  - begin()
  - end()
  - size()
  - empty()
  - clear()
  - erase()
  - insert()
  - … C++ reference

vector

- insert(), erase()

```
1  #include <vector>
2
3  int a[5] = {0, 1, 3, 4, 5};
4  vector<int> v(a, a + 5); // 0, 1, 3, 4, 5
5
6  v.insert(v.begin() + 2, 2); // 0, 1, 2, 3, 4, 5
7  v.erase(v.begin() + 5); // 0, 1, 2, 3, 4
```

vector

- Multi-dimension

```cpp
#include <vector>

vector<int> v[2];

v[0].push_back(0);
v[1].push_back(1);
v[1].push_back(2);

for (int i = 0; i < v.size(); ++i) {
    for (int j = 0; j < v[i].size(); ++j) {
        printf("%d ", v[i][j]);
    }
    putchar('\n');
}
```

0	
1	2

made by electron & free999 & riljian

Outline

vector

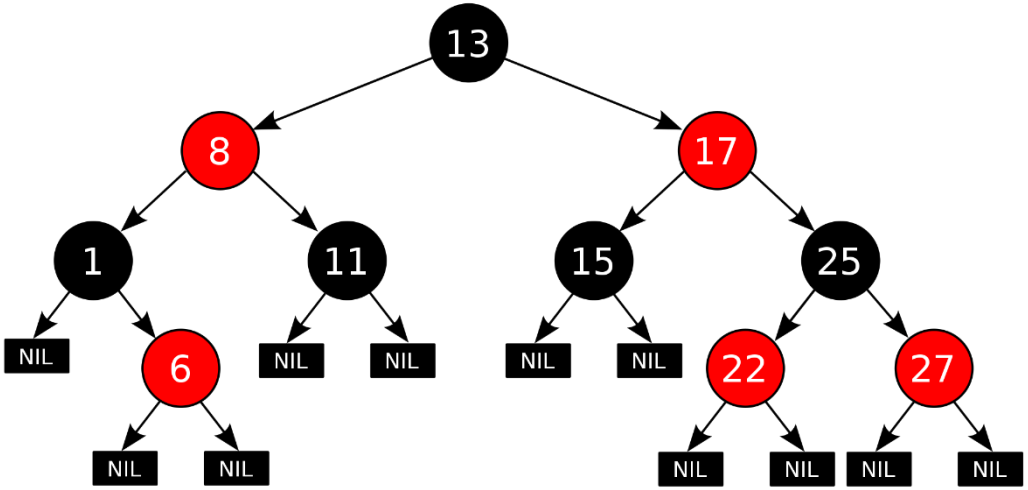
map

Hash

made by electron & free999 & riljian

map

- When
  - Key – Value
  - Wide range
    - 10000 numbers in 2^30 ~ 2^31 – 1
  - …
- Red-black tree
- Ordered



made by electron & free999 & riljian

map

- Useful member functions
  - begin()
  - end()
  - size()
  - empty()
  - operator[]
  - clear()
  - erase()
  - find()
  - … C++ reference

map

- operator[], find(), erase(), begin(), end()

```cpp
#include <map>

map<char, int> m;
map<char, int>::iterator it;

m['a'] = 50;
m['b'] = 100;
m['c'] = 150;

it = m.find('b');
if (it != m.end()) {
    m.erase(it);
}

for (it = m.begin(); it != m.end(); ++it) {
    printf("%c => %d\n", it->first, it->second);
}
```


made by electron & free999 & riljian

Outline

vector

map

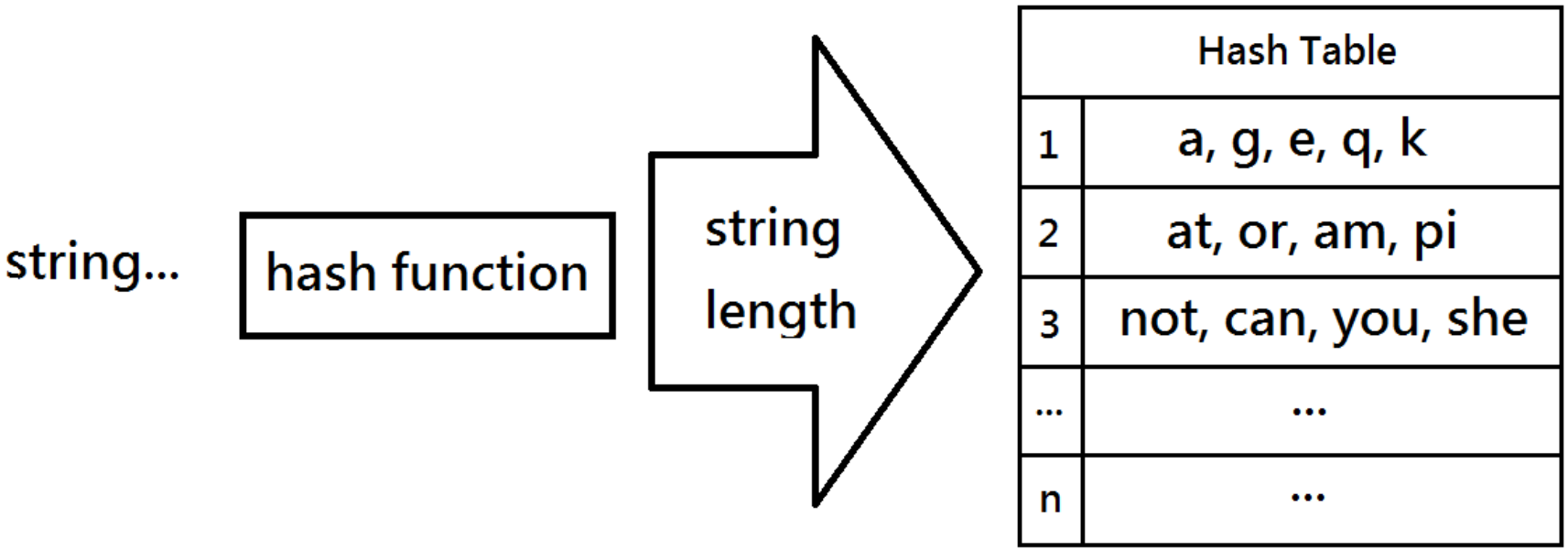
Hash

made by electron & free999 & riljian

Hash

- Quickly locate a data record

Hash



made by electron & free999 & riljian

Hash

- map + vector → hash table

```
1   #include <vector>
2   #include <map>
3
4   int hash_function(Object);
5
6   int address;
7   Object object;
8   map<int, vector<Object> > hash_table;
9
10  address = hash_function(object); // find the address of bucket
11  hash_table[address].push_back(object); // put the object into the bucket
```

practice

- vector
  - UVa 10895
- map
  - POJ 2503
  - UVa 10391
- hash
  - UVa 642
  - UVa 11991

made by electron & free999 & riljian