

# MCNL 여름 방학 스터디 2주차

**MCNL**

Yunmin Go

School of CSEE



# Schedule

- 1주차: Segment tree, Union-Find, Red-black tree, Trie
  - 2주차: Trie 기반 자동 완성, Red-Black Tree 기반 Map 구현
  - 3주차: Malloc 구현
  - 4주차: Network Programming study, Simple Search Engine 구현
  - 5주차: Network Programming study, Simple FTP 구현
  - 6주차: Simple Git Server 구현
  - Final Project: P2P 구현
- 
- 코딩 테스트 1주일에 3개씩

# Red-Black Tree 기반 Map 구현

- Red-Black Tree를 기반으로 C++ Map Class를 구현하시오.
  - map.cpp에서 사용한 STL map을 대체하는 my\_map class를 구현

```
#include <iostream>
#include <map>
#include <string>

using namespace std;

void print_map(map<string, int> m)
{
    map<string, int>::iterator iter;

    for (iter = m.begin(); iter != m.end(); iter++) {
        cout << iter->first << ": " << iter->second << '\n';
    }
}

int main()
{
    map<string, int> m;

    cout << "** First Step **\n";
    m.insert(make_pair("Global", 10));
    m.insert(make_pair("Handong", 30));
    m.insert(make_pair("CSEE", 20));
    m.insert(make_pair("MCNL", 15));
    print_map(m);
```

```
    cout << "\n** Second Step **\n";
    m["Pohang"] = 50;
    m["Korea"] = 60;
    print_map(m);

    cout << "\n** Third Step **\n";
    m["CSEE"] = 100;
    m.erase("Global");
    print_map(m);

    cout << "\n** Fourth Step **\n";
    string key = "MCNL";
    if (m.find(key) != m.end()) {
        cout << key << " Exists! \n";
    } else {
        cout << key << " does not exist! \n";
    }

    cout << "\n** Fifth Step **\n";
    key = "Yunmin";
    if (m.find(key) != m.end()) {
        cout << key << " Exists! \n";
    } else {
        cout << key << " does not exist! \n";
    }

    return 0;
}
```

# Red-Black Tree 기반 Map 구현

- Red-Black Tree를 기반으로 C++ Map Class를 구현하시오.
  - My\_map class는 main.cpp에서 필요한 기능들만 제공하면 됨
    - STL map 과 같이 다양한 function 들을 구현할 필요 없음
  - 구현해야 하는 사항은 아래와 같음
    - **Class template**: key, value type에 대한 generic class를 정의
    - insert(): key, value 삽입
    - erase(): key 삭제
    - find(): key 검색
    - begin(): 시작 iterator
    - end(): 끝 iterator
    - [ ] operator: key로 map에 접근 → **Operator overloading**
    - **iterator**: first(=key), second(=value)
  - \* 자세한 정의는 <https://cplusplus.com/reference/map/map/> 참고  
(STL map과 굳이 똑같은 정의를 사용할 필요는 없음)

# Red-Black Tree 기반 Map 구현

- Red-Black Tree를 기반으로 C++ Map Class를 구현하시오.
  - STL map 소스 코드 (gcc-4.6.2)

[https://gcc.gnu.org/onlinedocs/gcc-4.6.2/libstdc++/api/a01067\\_source.html](https://gcc.gnu.org/onlinedocs/gcc-4.6.2/libstdc++/api/a01067_source.html)