Problem Link:

<https://leetcode.com/problems/design-movie-rental-system/description/?envType=daily-question&envId=2025-09-21>

Solution:

class MovieRentingSystem {

map<pair<int, int>, int> p;

map<int, set<pair<int, int>>> a;

set<tuple<int, int, int>> r;

public:

MovieRentingSystem(int n, vector<vector<int>>& e) {

for(auto& x : e)

{

int s = x[0], m = x[1], c = x[2];

p[{s, m}] = c;

a[m].insert({c, s});

}

}

vector<int> search(int m) {

vector<int> v;

for(auto& [c, s] : a[m])

{

v.push\_back(s);

if(v.size() == 5)

break;

}

return v;

}

void rent(int s, int m) {

int c = p[{s, m}];

a[m].erase({c, s});

r.insert({c, s, m});

}

void drop(int s, int m) {

int c = p[{s, m}];

r.erase({c, s, m});

a[m].insert({c, s});

}

vector<vector<int>> report() {

vector<vector<int>> v;

for(auto& [c, s, m] : r)

{

v.push\_back({s, m});

if(v.size() == 5)

break;

}

return v;

}

};

/\*\*

\* Your MovieRentingSystem object will be instantiated and called as such:

\* MovieRentingSystem\* obj = new MovieRentingSystem(n, entries);

\* vector<int> param\_1 = obj->search(movie);

\* obj->rent(shop,movie);

\* obj->drop(shop,movie);

\* vector<vector<int>> param\_4 = obj->report();

\*/