Problem Link:

<https://leetcode.com/problems/design-task-manager/description/?envType=daily-question&envId=2025-09-18>

Solution:

class TaskManager {

private:

map<int, pair<int, int>> m;

priority\_queue<pair<int, int>> h;

public:

TaskManager(vector<vector<int>>& tasks) {

for(const auto& x : tasks)

{

int u = x[0], t = x[1], p = x[2];

m[t] = {p, u};

h.emplace(p, t);

}

}

void add(int u, int t, int p) {

m[t] = {p, u};

h.emplace(p, t);

}

void edit(int t, int p) {

if(m.count(t))

{

int u = m[t].second;

m[t] = {p, u};

h.emplace(p, t);

}

}

void rmv(int t) {

m.erase(t);

}

int execTop() {

while(!h.empty())

{

auto [p, t] = h.top();

h.pop();

if(m.count(t) && m[t].first == p)

{

int u = m[t].second;

m.erase(t);

return u;

}

}

return -1;

}

};

/\*\*

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

\* TaskManager\* obj = new TaskManager(tasks);

\* obj->add(userId,taskId,priority);

\* obj->edit(taskId,newPriority);

\* obj->rmv(taskId);

\* int param\_4 = obj->execTop();

\*/