Problem Link:

<https://leetcode.com/problems/design-a-food-rating-system/description/?envType=daily-question&envId=2025-09-17>

Solution:

class FoodRatings {

private:

unordered\_map<string, pair<string, int>> fi;

unordered\_map<string, set<pair<int, string>>> cf;

public:

FoodRatings(vector<string>& foods, vector<string>& cuisines, vector<int>& ratings) {

for(int i = 0; i < foods.size(); ++i)

{

string& food = foods[i];

string& cuisine = cuisines[i];

int rating = ratings[i];

fi[food] = {cuisine, rating};

cf[cuisine].insert({-rating, food});

}

}

void changeRating(string food, int newRating) {

auto& [cuisine, oldRating] = fi[food];

auto& foodSet = cf[cuisine];

foodSet.erase({-oldRating, food});

foodSet.insert({-newRating, food});

fi[food].second = newRating;

}

string highestRated(string cuisine) {

return cf[cuisine].begin()->second;

}

};

/\*\*

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

\* FoodRatings\* obj = new FoodRatings(foods, cuisines, ratings);

\* obj->changeRating(food,newRating);

\* string param\_2 = obj->highestRated(cuisine);

\*/