<https://leetcode.com/problems/design-underground-system/>

var UndergroundSystem = function() {

this.travel = {}

this.travelTime = {}

};

/\*\*

\* @param {number} id

\* @param {string} stationName

\* @param {number} t

\* @return {void}

\*/

UndergroundSystem.prototype.checkIn = function(id, stationName, t) {

if(!this.travel.hasOwnProperty(id)){

this.travel[id] = {}

this.travel[id]["startStationName"] = stationName

this.travel[id]["startTime"] = t

}

};

/\*\*

\* @param {number} id

\* @param {string} stationName

\* @param {number} t

\* @return {void}

\*/

UndergroundSystem.prototype.checkOut = function(id, stationName, t) {

//StartTime should be smaller than endTime

if(this.travel.hasOwnProperty(id) && this.travel[id]["startTime"] < t){

this.travel[id]["endStationName"] = stationName

this.travel[id]["endTime"] = t

//this.travelTime[this.travel[id]["startStationName"]] = {}

//Checking exisiting entry

if(this.travelTime.hasOwnProperty(this.travel[id]["startStationName"]) && this.travelTime[this.travel[id]["startStationName"]].hasOwnProperty(stationName)){

this.travelTime[this.travel[id]["startStationName"]][stationName]["totalTime"] +=

this.travel[id]["endTime"] - this.travel[id]["startTime"]

this.travelTime[this.travel[id]["startStationName"]][stationName]["noOfTravel"] += 1

}else{

if(!this.travelTime.hasOwnProperty(this.travel[id]["startStationName"])){

this.travelTime[this.travel[id]["startStationName"]] = {}

}

this.travelTime[this.travel[id]["startStationName"]][stationName] = {}

this.travelTime[this.travel[id]["startStationName"]][stationName]["totalTime"] =

this.travel[id]["endTime"] - this.travel[id]["startTime"]

this.travelTime[this.travel[id]["startStationName"]][stationName]["noOfTravel"] = 1

}

// delete this.travel[id]

}

};

/\*\*

\* @param {string} startStation

\* @param {string} endStation

\* @return {number}

\*/

UndergroundSystem.prototype.getAverageTime = function(startStation, endStation) {

//console.log(this.travelTime[startStation],this.travelTime[startStation][endStation])

//console.log(this.travel,this.travelTime)

if(this.travelTime.hasOwnProperty(startStation) && this.travelTime[startStation].hasOwnProperty(endStation)){

let totalTravelTime = this.travelTime[startStation][endStation]["totalTime"]

let totalTravel = this.travelTime[startStation][endStation]["noOfTravel"]

return totalTravelTime/totalTravel

}else{

return 0

}

};

/\*\*

checkIn

We need to verify if user has not already checkendin

checkOut

45 : {

startStation,

startTime:

endStation:

endTime:

}

travelTime :{

startStation:{

endStation:{

travelTime:[7,9]

totalTime:

noOfTrave:

avg:

}

endStation:[]

}

}

getAverageTime

\*/

/\*\*

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

\* UndergroundSystem obj = new UndergroundSystem();

\* obj.checkIn(id,stationName,t);

\* obj.checkOut(id,stationName,t);

\* double param\_3 = obj.getAverageTime(startStation,endStation);

\*/

/\*\*

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

\* var obj = new UndergroundSystem()

\* obj.checkIn(id,stationName,t)

\* obj.checkOut(id,stationName,t)

\* var param\_3 = obj.getAverageTime(startStation,endStation)

\*/