Data Structures
Final Project

2016/12/15 10:00

Deadline
2017/01/05 10:00

UBikeSystem

- The target of the final project is to finish the implementation of a new complete system for the U-bike company.

Rental MRT Stores

- 12 rental stores of U-bike.

Danshui(淡水)	Hongshulin(紅樹林)	Beitou(北投)	Shilin(士林)
Zhongshan(中山)	Xinpu(新埔)	Ximen(西門)	Liuzhangli(六張犁)
Muzha(淡水)	Guting(淡水)	Gongguan(公館)	Jingmei(景美)

- The discount distance of each road between 2 stations is given in an input file.

Format	**“Station A” “Station B” “Discount distance of the road AB”**
Example	Danshui Hongshulin 49 Danshui Beitou 58 Hongshulin Beitou 1

Rental Charges

- We offer a discount to those who drive within shortest discount distance path between the stations where you rent and return the bike.

Rate Table	Electric	Lady	Road	Hybrid
Discount (≤)	$25/mile	$20/mile	$10/mile	$15/mile
Original (>)	$40/mile	$30/mile	$20/mile	$25/mile

U-Bike

```
bool isRented
```

```
int mileage
```

```
int heapIndex
```

```
std::string license
```

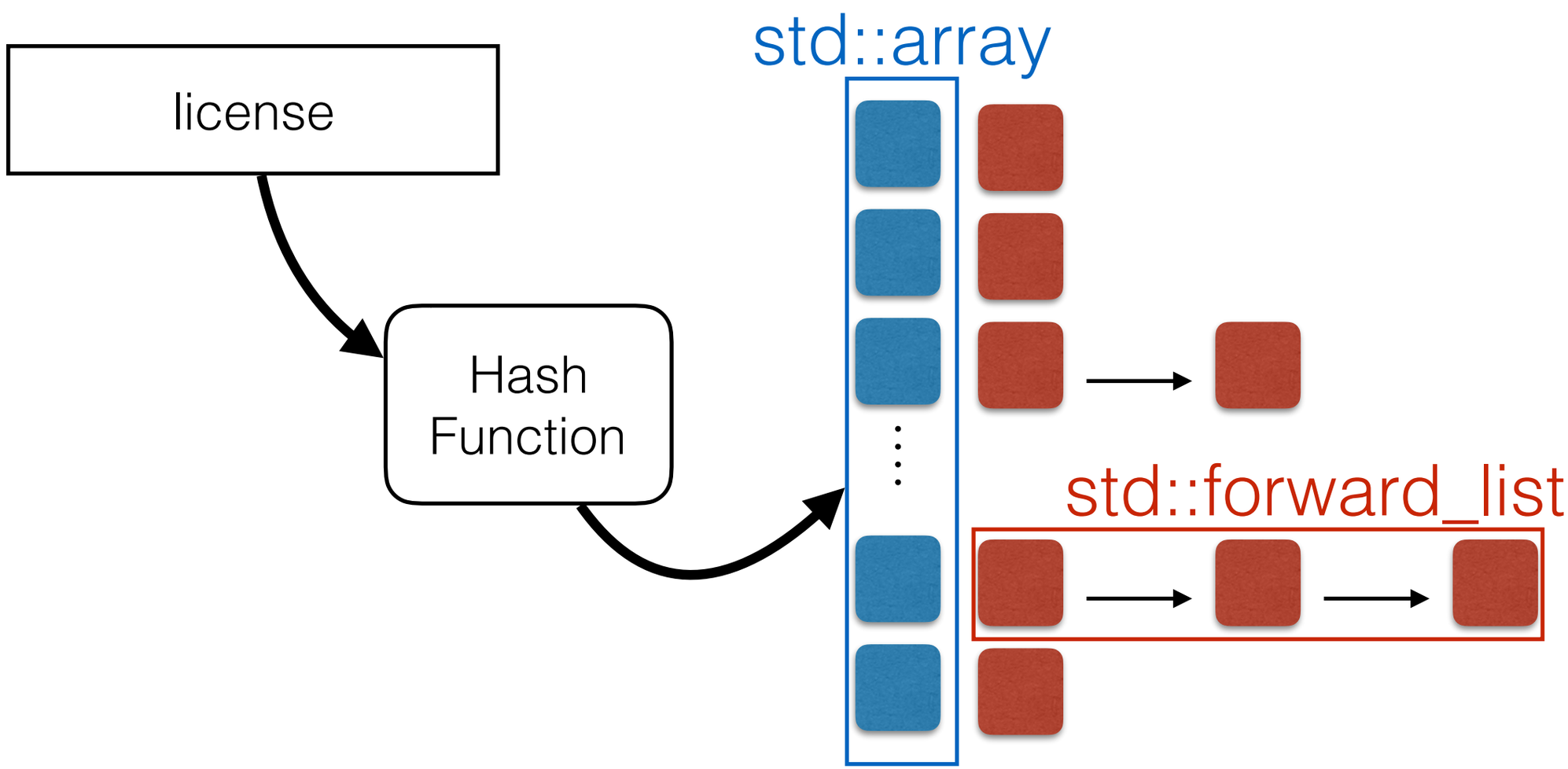
```
std::string station
```

```
std::string classType
```

```
//================================================
// if this ubike is rented
//================================================
bool isRented;

//================================================
// the mileage of this ubike
//================================================
int mileage;

//================================================
// the heap index of the heap where the ubike is.
//================================================
int heapIndex;

//================================================
// the license of this bike
//================================================
std::string license;

//================================================
// the station to which this ubike belongs
// it can be any string in StationNames
// "Danshui", "Hongshulin", "Beitou", ...
//================================================
std::string station;

//================================================
// the class type of this ubike
// it can be any string in HeapNames
// "Electric", "Lady", "Road" ...
//================================================
std::string classType;
```

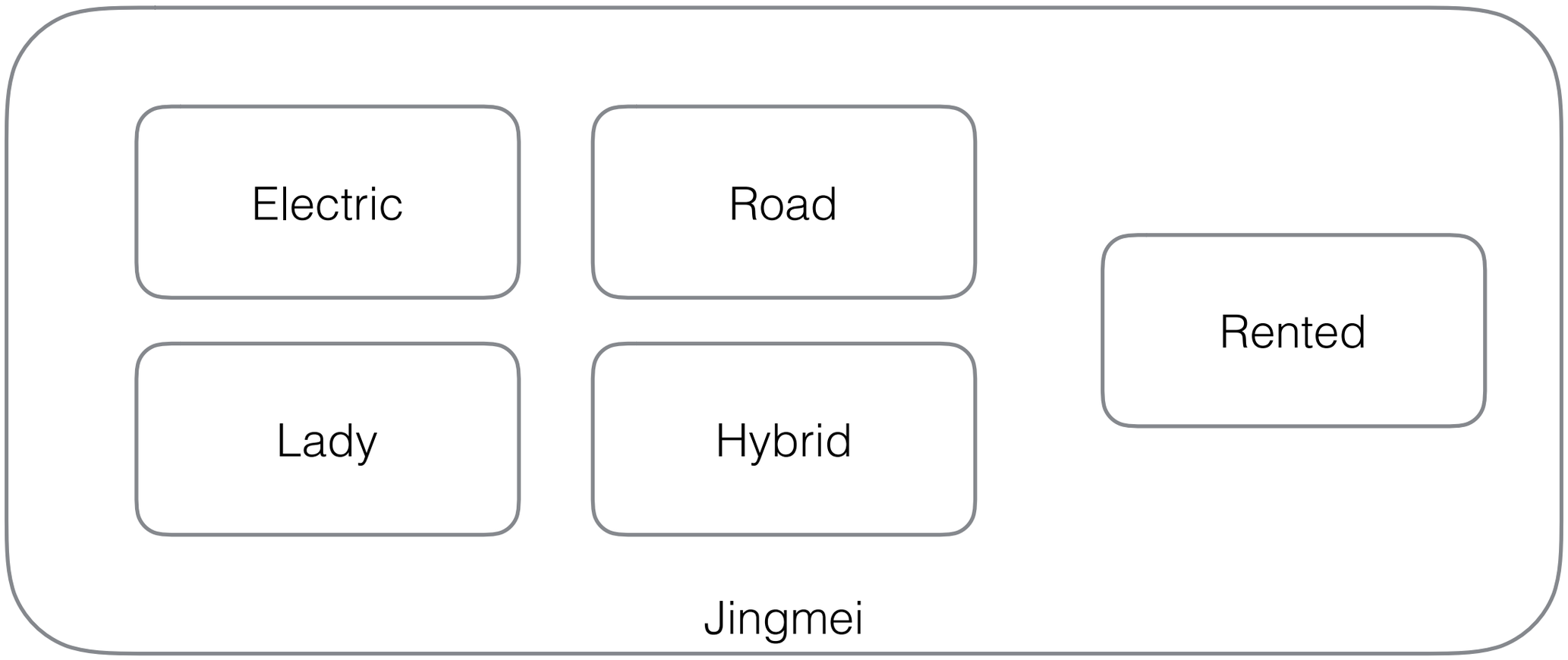
Hash table

- In order to locate bikes quickly by only providing license tag (5 alphanumeric characters A..Z and 0..9), a hash table is used.



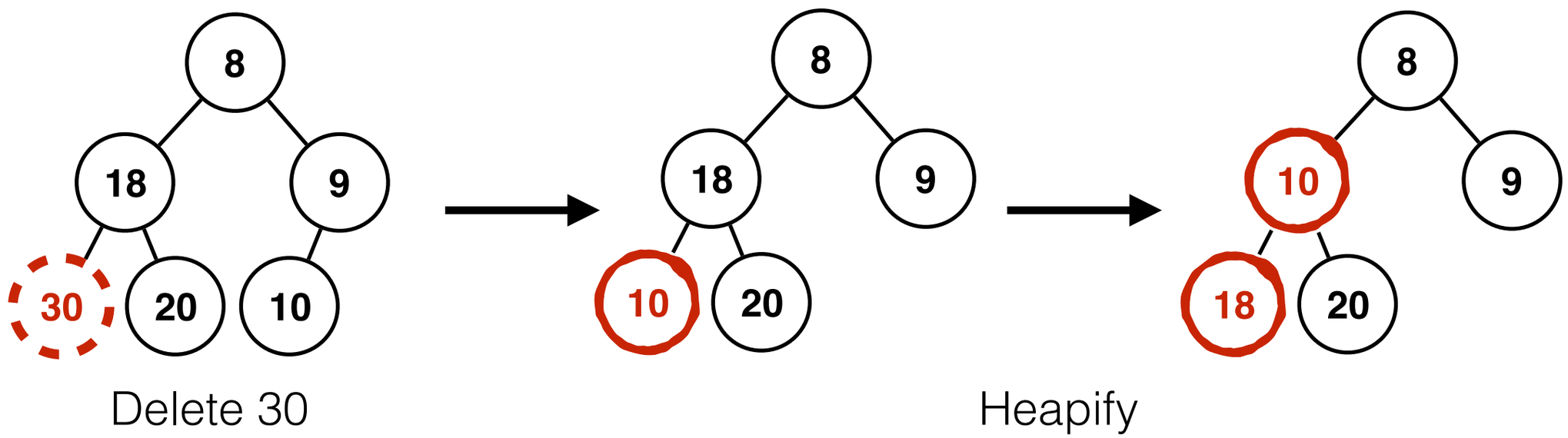
Inside a Store

- There are 4 min heaps for the four types of bikes and a min heap for rented U-bikes.



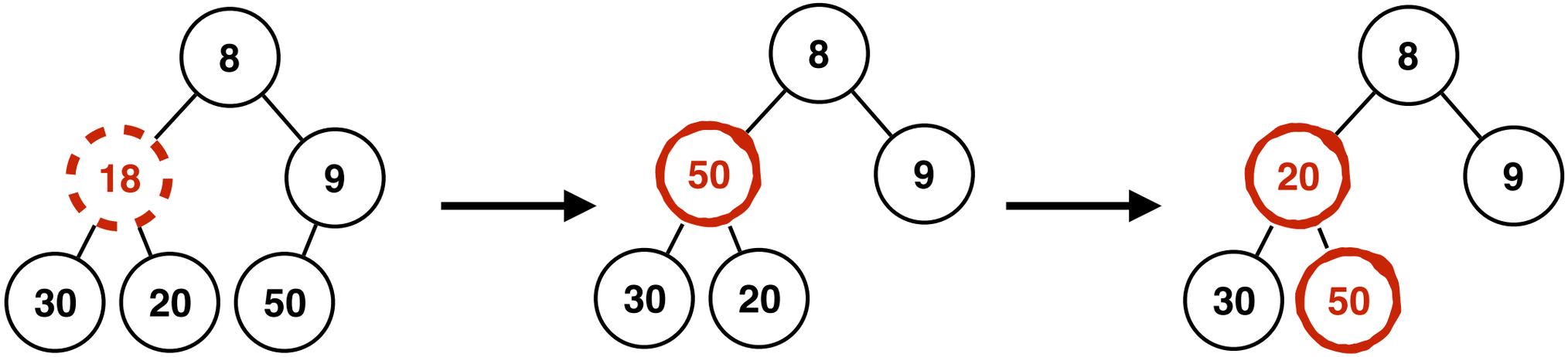
Heap

- Customized deletion - Bubble up

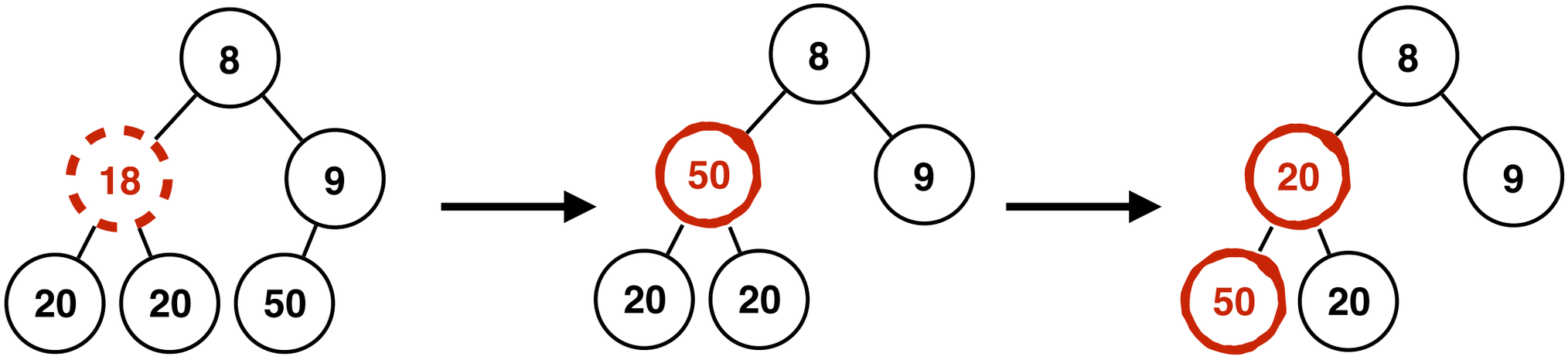


Heap

- Customized deletion - Bubble down



Delete 18 Heapify



Delete 18 Heapify

Test Case

- In each test case
  - cm.in - contains a series of commands.
  - map.in - the pairwise distance between stations.
  - ans.out - the system status after the execution of the commands.

Input Commands

- The input file cm.in consists of the following 5 kinds of commands, each of which implies calling the corresponding function of UBikeSystem.

1	**NewBike classType license mile station** Call UBikeSystemADT::NewBike(classType, license, mile, station);
2	**JunkIt license** Call UBikeSystemADT::JunkIt(license);
3	**Rent station classType** Call UBikeSystemADT::Rent(classType, station);
4	**Returns station license returnMile (total current mileage)** Call UBikeSystemADT::Return(station, license, returnMile);
5	**Trans station license** Call UBikeSystemADT::Trans(station, license);

Evaluation

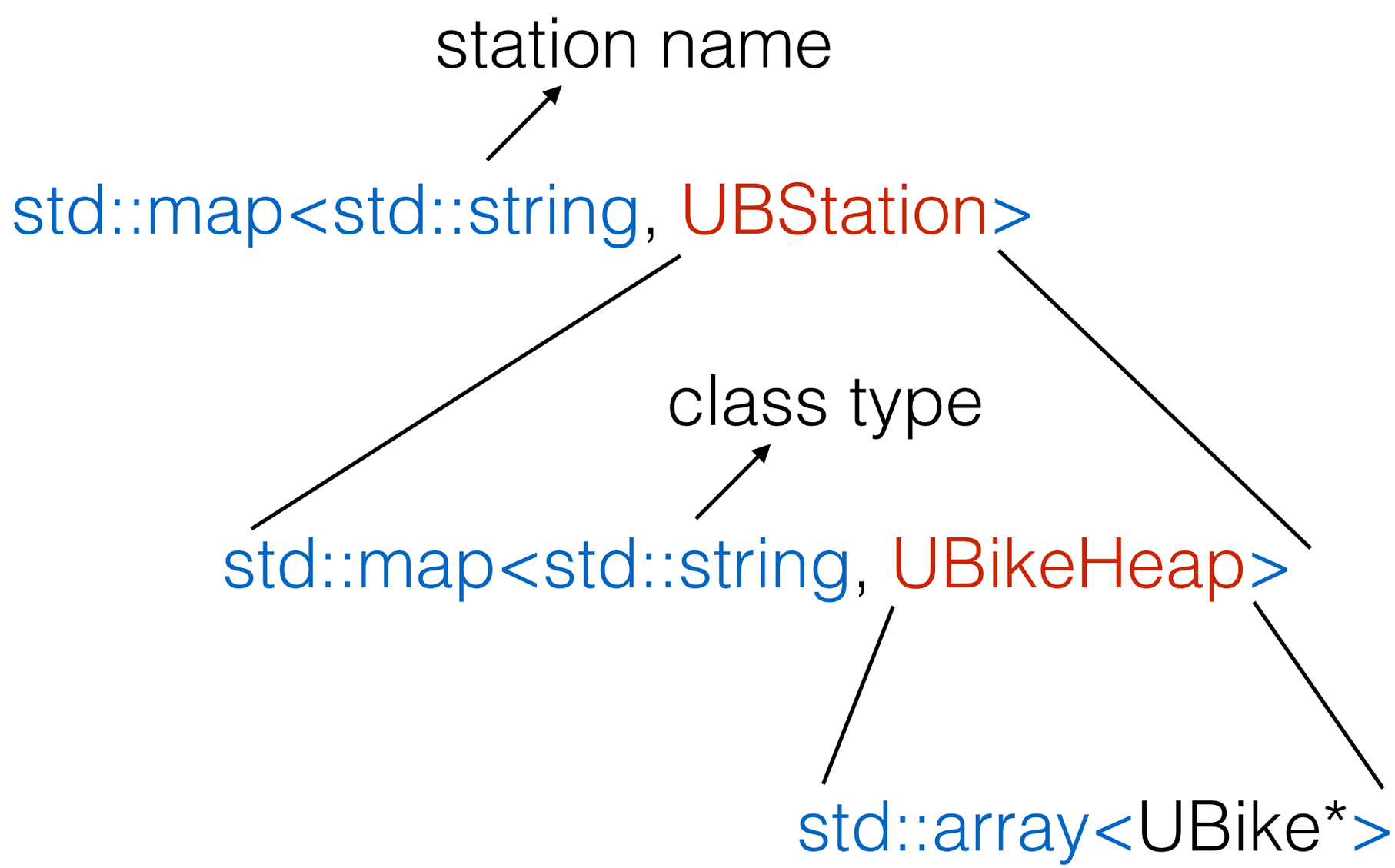
- For each test case, the evaluation is done by comparing the final status of UBikeSystem with the correct answer (ans.out).

```
//read the answer texts
std::ifstream ifsAns( AnsFile );
std::string ansText( (std::istreambuf_iterator<char>(ifsAns)),
                     (std::istreambuf_iterator<char>()      ) );
ifsAns.close();

//your output
std::ostringstream oss;
oss << ubSystem.toString();

//…

//compare the results
if( oss.str() == ansText )
    std::cout << "PASS!" << std::endl;
else
    std::cout << "FAIL." << std::endl;
```

Data Structure

- ConstParams.h: constant parameters.
- UBike: Abstracted UBike.
- PriceTable: To calculate the distance and the profit.
- UBikeHashTable: The hash table.
- UBikeHeap: The customized heap.
- UBikeSystem: The whole system that process every command in the input test cases.

UBikeSystem::ubStations



Submission

- The **team leader** is asked to submit to **ILMS** an archive, named **ds2016f_final.zip**
  - It should be compiled properly using **makefile**.
- Deadlines
  - 2016/12/29: Pre-Evaluation
  - 2017/01/05: Deadline