

# 110062272 盧思樺

## 1. How to compile and execute your program and give an execution example.

在 Linux terminal 的 DS\_final\_project 資料夾路徑下，輸入 make

case=case1 version=basic，就能執行。

```
hua@LAPTOP-58PMDL7:/mnt/c/Users/user/DS_final_project$ make case=case1 version=basic
g++ -g -std=c++11 -o ./bin/main ./src/*.cpp
./bin/main case1 basic
You have set case1 as your testcase:
running basic currently
-----
start your basic version of data structure final from here!
-----
finished computation at Sun Jan  8 14:51:16 2023
elapsed time: 0.0222204s
```

## 2.The details of your data structures. What data structures did you use, and how did you implement those data structures.

### (1) vector

usage:模仿 stl 中的 vector，讓我能夠動態儲存資料。

```
template<typename T>
class vector
{
public:
    vector():size(0),capacity(5),array(new T[capacity]) {}
    vector(const vector& rhs):size(rhs.size),capacity(rhs.capacity+5),array(new T[rhs.capacity+5])
    {
        for(int i=0;i<rhs.Size();i++)
        {
            array[i]=rhs.array[i];
        }
    }
    vector(int elements,int value=0):size(elements),capacity(elements+5),array(new T[capacity])
    {
        for(int i=0;i<size;i++)
        {
            array[i]=value;
        }
    }
    void push_back(const T value);
    void pop_back();
    bool empty() const;
```

```

    bool empty() const;
    int Size() const;
    int Capacity() const;
    T& operator[](int index);
    //vector<T>& operator=(const vector<T>& rhs);
    T& front();
    T& back();
    void clear();
private:
    int size;
    int capacity;
    T* array;
};

template<typename T>
bool vector<T>::empty() const
{
    return size==0;
}

```

```

template<typename T>
int vector<T>::Size() const
{
    return size;
}

template<typename T>
int vector<T>::Capacity() const
{
    return capacity;
}

template<typename T>
void vector<T>::push_back(T value)
{
    if(size<capacity)
    {
        array[size++]=value;
    }
}

```

```

    }
    else
    {
        capacity*=2;
        T* temp=new T[capacity];
        for(int i=0;i<size;i++)
        {
            temp[i]=array[i];
        }
        temp[size++]=value;
        //delete[] array;
        array=temp;
    }
}

```

```

template<typename T>
int vector<T>::Size() const
{
    return size;
}

template<typename T>
int vector<T>::Capacity() const
{
    return capacity;
}

template<typename T>
void vector<T>::push_back(T value)
{
    if(size<capacity)
    {
        array[size++]=value;
    }
}

```

```

    }
    else
    {
        capacity*=2;
        T* temp=new T[capacity];
        for(int i=0;i<size;i++)
        {
            temp[i]=array[i];
        }
        temp[size++]=value;
        //delete[] array;
        array=temp;
    }
}

```

```

    size--;
}

template<typename T>
T& vector<T>::front()
{
    return array[0];
}

template<typename T>
T& vector<T>::back()
{
    return array[size-1];
}

template<typename T>
T& vector<T>::operator[](int index)
{
    return array[index];
}

```

## (2) Bike

Usage:儲存任何有關於腳踏車的資訊。

## (3) User

Usage: 儲存任何有關於使用者的資訊。

## (4) record

Usage:用來儲存要輸入進 station\_status.txt 的資料。

## (5) Station

Usage:一個單純用來儲存站點的類別，跟建圖有關。

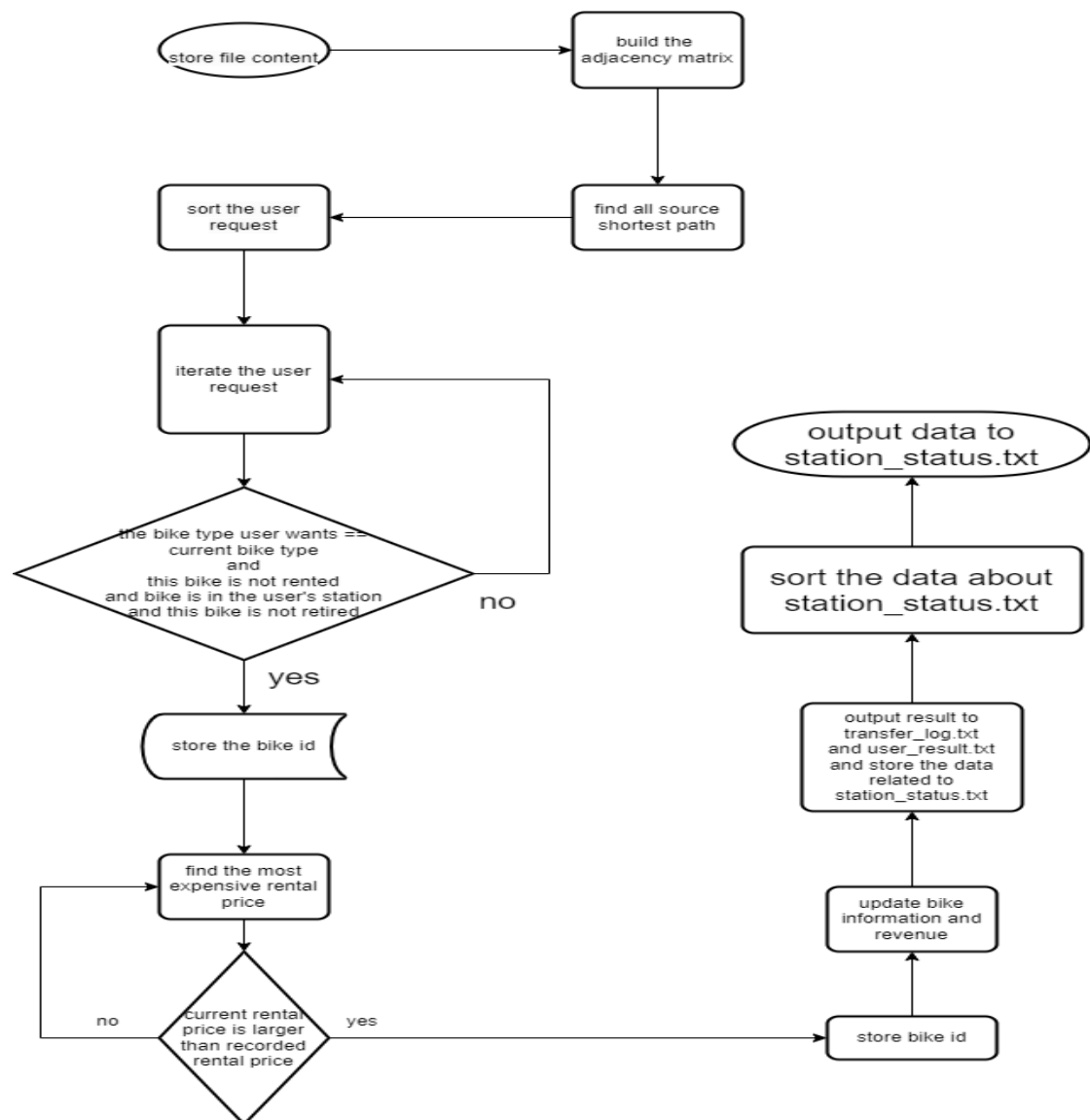
## (6) BikeType

Usage:用來儲存 bike\_info.txt 的資料，特別是紀錄 rental count limit 和

depreciation 的部分。

3. The details of your algorithm. You could use flow chart(s) and/or pseudo code to help elaborate your algorithm.

利用 Floyd-Warshall' s Algorithm 去算出最短路徑，然後在把 user request 存下來做排序，然後跑遍所有的 user request，去找能符合每個 user 條件能拿到的最貴腳踏車。



4. [Optional] If you have any feedback, please write it here.

Such as comments for improving the spec of this assignment,  
etc.

希望做學生報告時，助教能提供範本給我們做參考，不然有點無所適從，不知道報告內容要講的多詳細，感謝助教。