Section 1. Sorting….. Interrupted.

For the following questions, you are a given a starting array of integers. The array is then sorted by **one of the sorting algorithm** discussed in this course (**Selection, Insertion, Bubblesort**).

However, due to a power outage, the sorting is **interrupted, i.e. the sorting is NOT complete.**

Based on the given partially sorted array, give the following:

i. Which sorting algorithm you think is used? Choose **exactly one** even if there may be multiple answers.

ii. Briefly explain **your choice,** i.e. how do you know it is that particular sorting algorithm.

1. Original Array:

24	76	85	56	95	34	37	79

Partially Sorted Array:

24	76	85	56	79	34	37	95

(3 marks)

2. Original Array:

45	28	62	69	28	38	56	53

Partially Sorted Array:

28	28	45	62	69	38	56	53

(3 marks)

Section 2. Sorting….. Reinvented.

3. Mr.Vidoc rewrote the bubble sort function to the following:

```
void BBS(int a[], int n) {
    int i, j;
    for (i = n-1; i >= 1; i--) {
        for (j = 1; j <= i; j++) {
            if ( outOfOrder( a[j], a[j-1] ) ) {
                swap(a, j-1, j);
            }
        }
    }
}
```

```
bool outOfOrder( int A, int B)
{
    if (A <= B)
        return true;
    return false;
}
```

Does the BBS() function sort the array a[] in *ascending* (small to large) or *descending* order (large to small)?

Briefly explain your answer.

(4 marks)

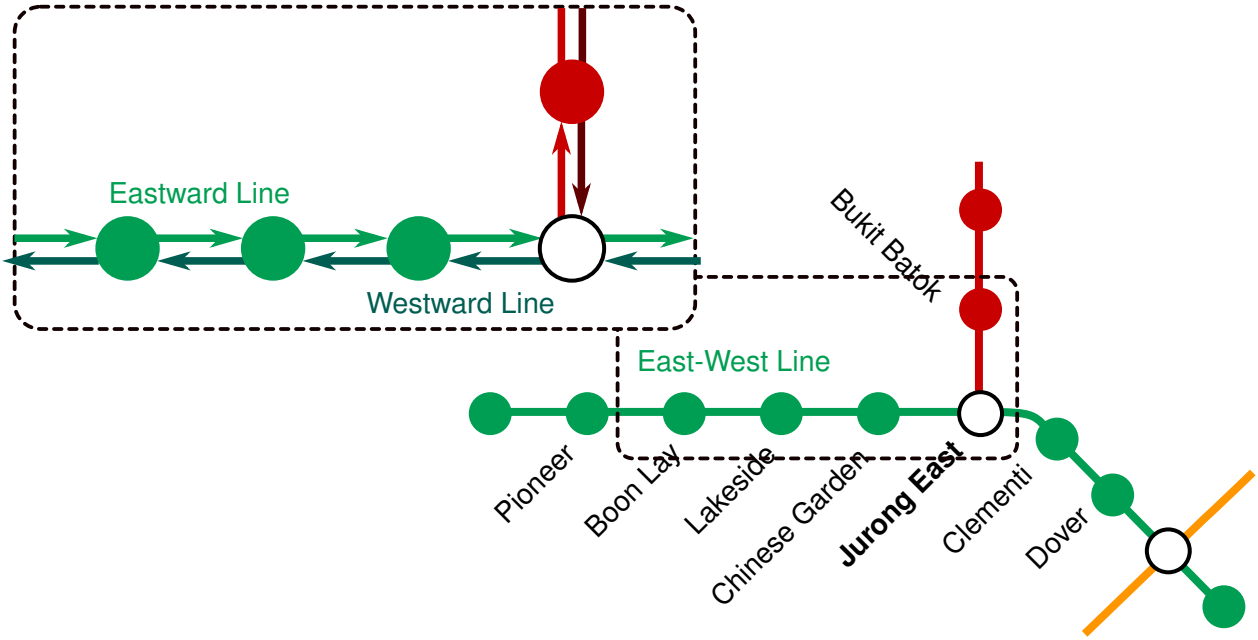
4. Refering to the same updated BBS() function written by Mr.Vidoc.

Is the BBS() stable?

Briefly explain.

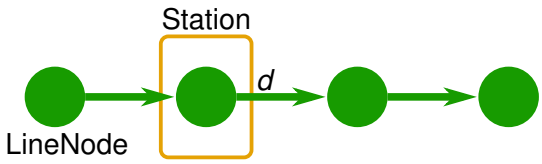
(4 marks)

Question 1: Subway Routes [12 marks]



A subway transit system consists of stations connected to each other by one or more train lines. Though a bi-directional line is usually thought of as one, in actual fact it behaves more like two uni-directional line.

Suppose we model a line as a linked list of `LineNode`.



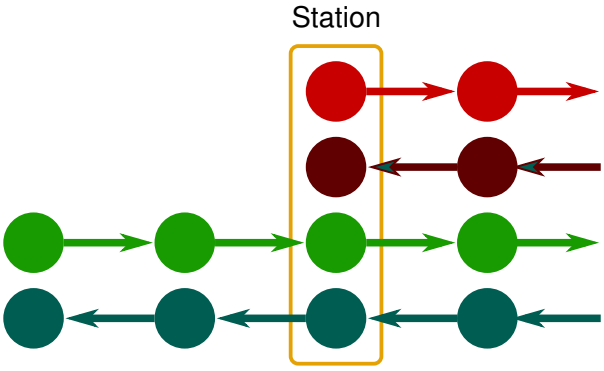
A `LineNode` is implemented as the following struct:

```
struct LineNode {
    Station *station;
    int distance;
    LineNode *next;
};
```

where `station` is a pointer to the current station this node is part of, `distance` is the distance to the next LineNode which is pointed to by `next`. The last node in the line will have the next pointer be `NULL`.

A. Implement the function `int distance_between(LineNode *node, Station &station)` which takes as input a LineNode and a Station, and returns the distance between them. If it is not possible to reach the station -1 is returned. [6 marks]

A `Station` thus consists of one or many lines:



A `Station` is implemented as the following struct:

```
struct Station {
    string name;
    vector<LineNode*> lines;
};
```

B. Implement the function `int distance_between(Station &a, Station &b)` which takes as inputs two `Station`, and returns the minimum distance between them on a single line.

Note that while no line changes take place, it is possible for a line to loop round on itself, like a continuous circle. [6 marks]

Question 2: Cars [4 marks]

Consider the following classes:

```
class Car {
protected:
    virtual int top_speed() {
        return 120;
    }

public:
    virtual void drive(int speed) {
        speed = min(speed, top_speed());
        cout << "Vroom! " << speed << " km/h" << endl;
    }
};

class SportsCar : public Car {
protected:
    bool nitro = false;

    int top_speed() {
        int t = Car::top_speed();
```

```
        if (nitro)
            t += 100;
        return t;
    }

public:
    virtual void toggle_nitro() {
        nitro = !nitro;
    }
};

class PoliceCar : public SportsCar {
    bool siren = false;

public:
    virtual void toggle_siren() {
        siren = !siren;
        nitro = nitro and siren;
    }

    virtual void toggle_nitro() {
        nitro = !nitro and siren;
    }

    void drive(int speed) {
        if (siren) {
            cout << "Bee Do Bee Do. ";
        }
        SportsCar::drive(speed);
    }
};
```

What is written to standard output when the following lines of code are executed:

```
PoliceCar pc;
pc.drive(200);
pc.toggle_nitro();
pc.drive(200);
pc.toggle_siren();
pc.drive(200);
pc.toggle_nitro();
pc.drive(200);
pc.toggle_siren();
pc.drive(200);
```

[4 marks]

— E N D O F Q U E S T I O N S —