📘 Exercise Set 11.2 – Solutions

🧮 Chapter 11: The World of Algorithms

1. 📋 Suppose List 1 and List 2 are two lists of numbers in increasing order.

(i) Write an algorithm to find elements in List 1 that are not present in List 2.

Algorithm:

1. Start with an empty list called **List 1-only**.
2. Take each element x of List 1, one by one, from left to right.
3. Check whether x is present in List 2.
4. If x is **not present** in List 2, add x to **List 1-only**.
5. Continue until all elements of List 1 have been checked.
6. Report **List 1-only**.

Answer:

The resulting list contains all the elements that are present in **List 1 but not in List 2**.

(ii) Write an algorithm to find elements in List 2 that are not present in List 1.

Algorithm:

1. Start with an empty list called **List 2-only**.
2. Take each element x of List 2, one by one, from left to right.
3. Check whether x is present in List 1.
4. If x is **not present** in List 1, add x to **List 2-only**.
5. Continue until all elements of List 2 have been checked.
6. Report **List 2-only**.

Answer:

The resulting list contains all the elements that are present in **List 2 but not in List 1**.

---

2. 🔢 Describe an algorithm to compute the Least Common Multiple (LCM) of two numbers.

Let the two numbers be **m** and **n**.

Algorithm:

1. Find the multiples of `m`:
   `m, 2m, 3m, 4m, ...`
2. Find the multiples of `n`:
   `n, 2n, 3n, 4n, ...`
3. Compare the two lists of multiples.
4. Find the **smallest number that occurs in both lists**.
5. Report this number as **LCM(m, n)**.

Example:

Find LCM of 6 and 8.

Multiples of 6:

```
6, 12, 18, 24, 30, ...
```

Multiples of 8:

```
8, 16, 24, 32, ...
```

The first common multiple is **24**.

Answer:

LCM(6, 8) = 24

---

3. 🔍 Divisors occur in pairs

Question: Divisors occur in pairs. For instance, the divisors of 18 are:

`(1, 18), (2, 9), (3, 6)`

(i) If we write out divisors in pairs, how many numbers do we have to examine between 1 and n to find all the divisors of n?

We only need to examine numbers up to √**n**.

Why?

If `d` is a divisor of `n`, then its paired divisor is:

`n/d`

For example, for 18:

- `1 × 18`
- `2 × 9`
- `3 × 6`

Once we reach √18, the divisor pairs have been completely identified.

Answer:

We need to examine only the numbers from:

1 to √n

So, approximately √**n numbers** need to be examined.

(ii) If we list out the divisors in pairs, will our gcd algorithm still work in the manner we have described?

Yes, the gcd algorithm will still work.

The reason is that listing divisors in pairs still gives us **all the divisors** of the number.

For example, for 18:

`(1,18), (2,9), (3,6)`

These pairs contain all the divisors:

`1, 2, 3, 6, 9, 18`

We can then compare the complete divisor lists of the two numbers and identify their common divisors.

Answer:

Yes. The gcd algorithm will still work, provided that all divisor pairs are included when forming the complete divisor list.