the basic operation of GPIO in an

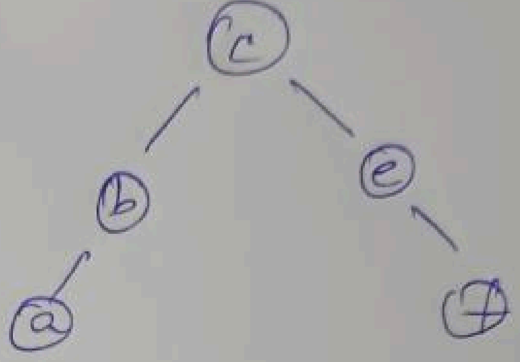
GPIO(General Purpose

that allo

Illustration:

P = [a, b, c, d, e]

F = [b, d, e, f, g]



2> Password Matching:

Harish. J
CB.EN.U4CCE23016

n password protected files with unique pwds.
F[i] → files P[i] → pwd.

Logic:

- To solve this problem, we could use a binary search tree. We would build this binary search tree according to the constraint by matching the ~~first~~ ~~to~~ file and its password.
- We would iterate through the list of ~~nodes~~ passwords and test them one by one. Upon recieving feedback (smaller or greater), we would manage the BST
- For each node, it will contain a password and a flag indicating whether it is matched.

Algorithm:

```
* Password matching [F, P]:
  1> Insert_BST(File array)
  2> Initialize ptr = root
  3> for i in range(0, length of P):
      3.1> While ptr ≠ Null:
          3.1.1> If ptr.password == P[i]:
                  • Print (Unlocked)
                  • Mark as flagged
                  • Delete (ptr). # to reduce complexity.
          3.1.2> If ptr.password > P[i]: # lexicography
                  • ptr = ptr.left
          3.1.3> Else, ~~ptr.right =~~ p
                  ptr = ptr.right
```