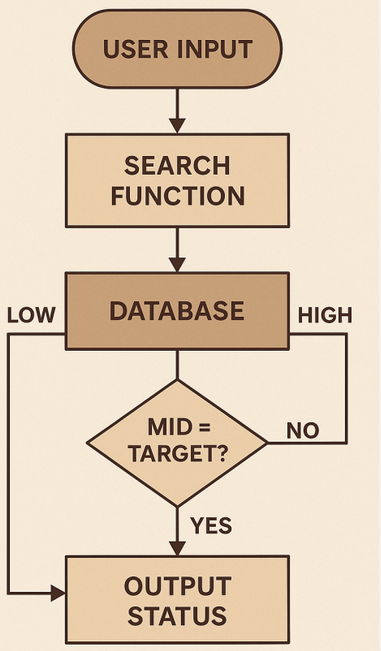
LOST AND FOUND

OPTIMIZING PARCEL SEARCH WITH BINARY BRILLIANCE

KIRTHIKA KS (RA2311043010082)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, KATTANKULATHUR.

FLOW CHART!!!!!



PROBLEM SUMMARY

A COURIER COMPANY MANAGES A LARGE SORTED LIST OF PARCELS BY TRACKING ID. CUSTOMERS OFTEN REQUEST REAL-TIME PARCEL UPDATES. THE SYSTEM NEEDS TO IDENTIFY PARCEL STATUS EFFICIENTLY WITH MINIMAL DELAY.

📦 PARCEL TRACKING ANALOGY

BINARY SEARCH IS LIKE GPS FOR PARCELS:
INSTEAD OF SEARCHING EVERY STREET (🔍 LINEAR SEARCH),
IT JUMPS HALFWAY EACH TIME (↔➡⬅),
ZEROING IN ON THE EXACT ADDRESS IN RECORD TIME.
📦 RESULT? CUSTOMER GETS PARCEL STATUS IN SECONDS.
🎯 EFFICIENT. PRECISE. FAST.

CHOSEN ALGORITHM: BINARY SEARCH

WE CHOSE BINARY SEARCH AS THE LIST IS ALREADY SORTED.
IT DIVIDES THE SEARCH SPACE IN HALF WITH EACH STEP, PROVIDING FASTER RESULTS COMPARED TO LINEAR SEARCH.

TIME & SPACE COMPLEXITY

- BINARY SEARCH:
  ⏳ TIME = O(LOG N)
  🧠 SPACE = O(1)

COMPARED TO

- LINEAR SEARCH:
  ⏳ TIME = O(N)

PSEUDOCODE

```
FUNCTION SEARCHPARCEL(TRACKINGLIST, TARGETID):
    LOW ← 0
    HIGH ← LENGTH(TRACKINGLIST) - 1

    WHILE LOW ≤ HIGH:
        MID ← (LOW + HIGH) // 2

        IF TRACKINGLIST[MID] == TARGETID:
            RETURN "PARCEL FOUND: DISPLAY STATUS"

        ELSE IF TRACKINGLIST[MID] < TARGETID:
            LOW ← MID + 1

        ELSE:
            HIGH ← MID - 1

    RETURN "PARCEL NOT FOUND"
```

🔗 OPEN CASE STUDY REPORT

REAL-WORLD RELEVANCE 🌍

- 📦 COMPANIES LIKE DHL, FEDEX, AMAZON USE OPTIMIZED SEARCH TO MANAGE MILLIONS OF PACKAGES.
- 🔁 BINARY SEARCH SPEEDS UP RETRIEVAL IN SORTED DATABASES.
- 📲 USER TYPES TRACKING ID ⁀ INSTANT PARCEL STATUS APPEARS.
- ⏱️ SAVES TIME, REDUCES SERVER LOAD, AND BOOSTS CUSTOMER SATISFACTION.

💡 EFFICIENCY IN SEARCHING = BETTER USER EXPERIENCE = HAPPY CUSTOMERS!

🔍 BEHIND THE SEARCH: WHY BINARY WINS

MOST REAL-WORLD SEARCH SYSTEMS — LIKE PARCEL TRACKING, STOCK INVENTORIES, OR EVEN YOUR FILE MANAGER — RELY ON SORTED DATA FOR A REASON.
WHEN THE LIST IS SORTED, BINARY SEARCH BECOMES THE HERO:
IT DIVIDES THE DATASET INTO HALVES REPEATEDLY UNTIL IT FINDS THE TARGET.
THAT'S LOGARITHMIC MAGIC: 1,000,000 ENTRIES? YOU'LL NEED JUST 20 STEPS 😳📉
COMPARE THAT TO LINEAR SEARCH, WHICH COULD TAKE ALL 1,000,000 STEPS! 🐢
BINARY SEARCH IS NOT ONLY FAST BUT ALSO MEMORY-EFFICIENT —
IT USES CONSTANT SPACE, MAKING IT IDEAL FOR SYSTEMS WITH LIMITED RESOURCES 📦💻
SO NEXT TIME YOUR DELIVERY SHOWS UP IN SECONDS AFTER ENTERING A TRACKING ID —
YOU KNOW THERE'S A CLEVER ALGORITHM WORKING BACKSTAGE. 🎭