Optimize Air Routes-

F = [1, 2000] [2, 4000] [3, 6000] | 7000

B = [1, 2000] [2, 2500] [3, 4000]

2000 — 2000, 2500, 4000

4000 — 2000, 2500, 4000

6000 — 2000, 2500, 4000.

```
optimise (forward, backward, target)
    max ← 0
    Arrays.sort (backward, natural ordering)

    for (int i ← 0 to forward.len - 1)
        c ← target - forward[i]
        index ← binarySearch (c)
        if (index != -1)
            sum ← forward[i] + back[index]
            if sum ≥ max
                if sum > max
                    result ← new()
                max ← max (sum, max)
                res.add (forward, backward)

    get result.
```

```
// return idx of
// max element ≤ t.

binarySearch (arr, t)
    lo ← 0  hi ← n-1
    while (lo ≤ hi)
        mid ← (lo+hi)/2
        if arr[mid] == target
            get mid
        if arr[mid] > target
            hi ← mid-1
        else
            lo ← mid+1
    get high.
```

high

example

4 [1 2 3 5 6]

5 [1 2 3 4 6]

8 [9] -1