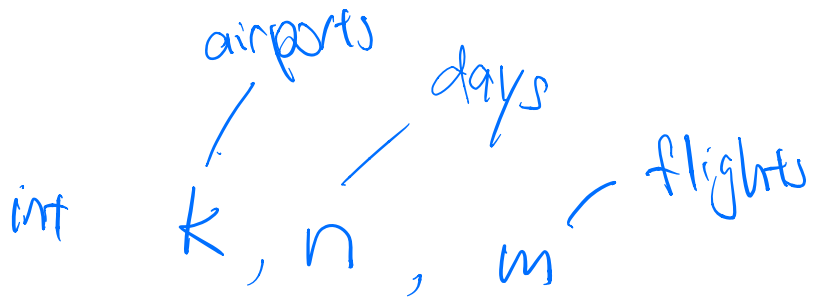
- generate their flight schedules & destinations
- Fill each flight schedule in the given window
  - Each customer can only take one flight a day
  - Can fly on any day on or after their suggested departure date
  - output optimal if all m flights can be filled to capacity



m lines: int u, v, d, z — day, capacity

kn lines: int a, b, c

a: airport, b: day, c: No. of customer

// Each day has an Adjacency List

vector < vector < vector < pair < int , int > > > Flights ;

// Arrival of customer at each airport at each day

int arrival [n] [k] ;

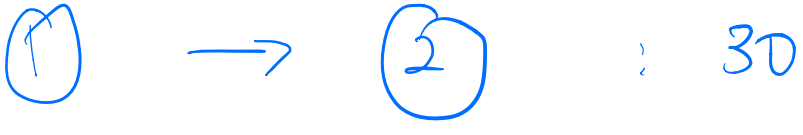
// At start of day, current no of customer at each airport

vector<int> current_cust ;

// At end of day, those customer who flew and will be added to next day's current_cust

vector<int> next_cust ;

Day 1



Day 2

② → ① : 50

Day 1:

②	10 ppl	②	40
①	30 ppl	①	0

Day 2

② 40 +	②	10 ppl	② 0
① 0 +	①	10 ppl	① 60