



# Philip J. Fry Problem

ANDREAS RONALDI - 6182101026





Explanation



## You are a captain

You are in a spaceship and need to make **n** trips for a mission,

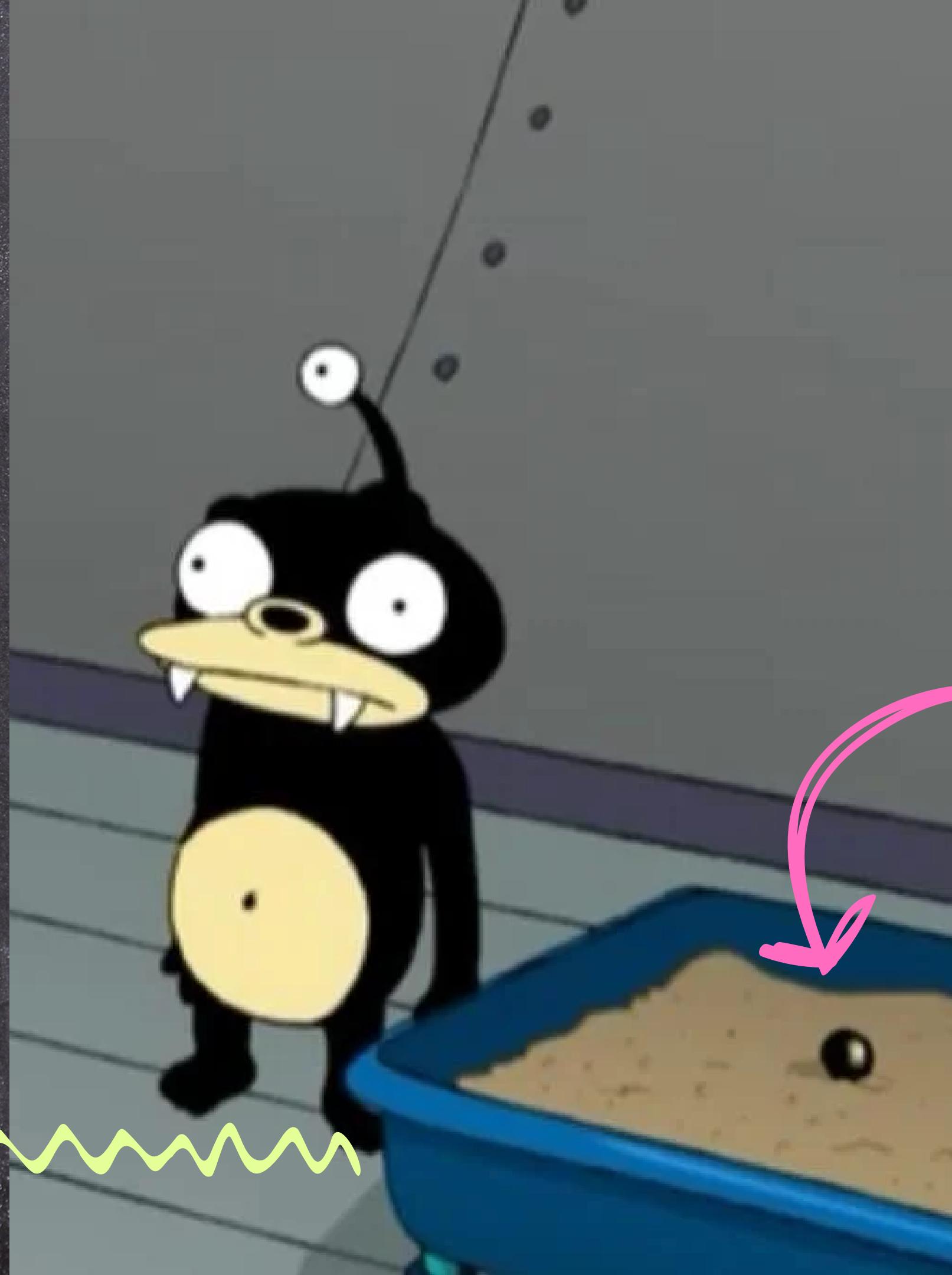
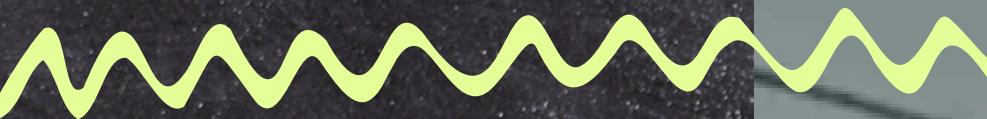
You are given a notebook how many minutes each trip will take

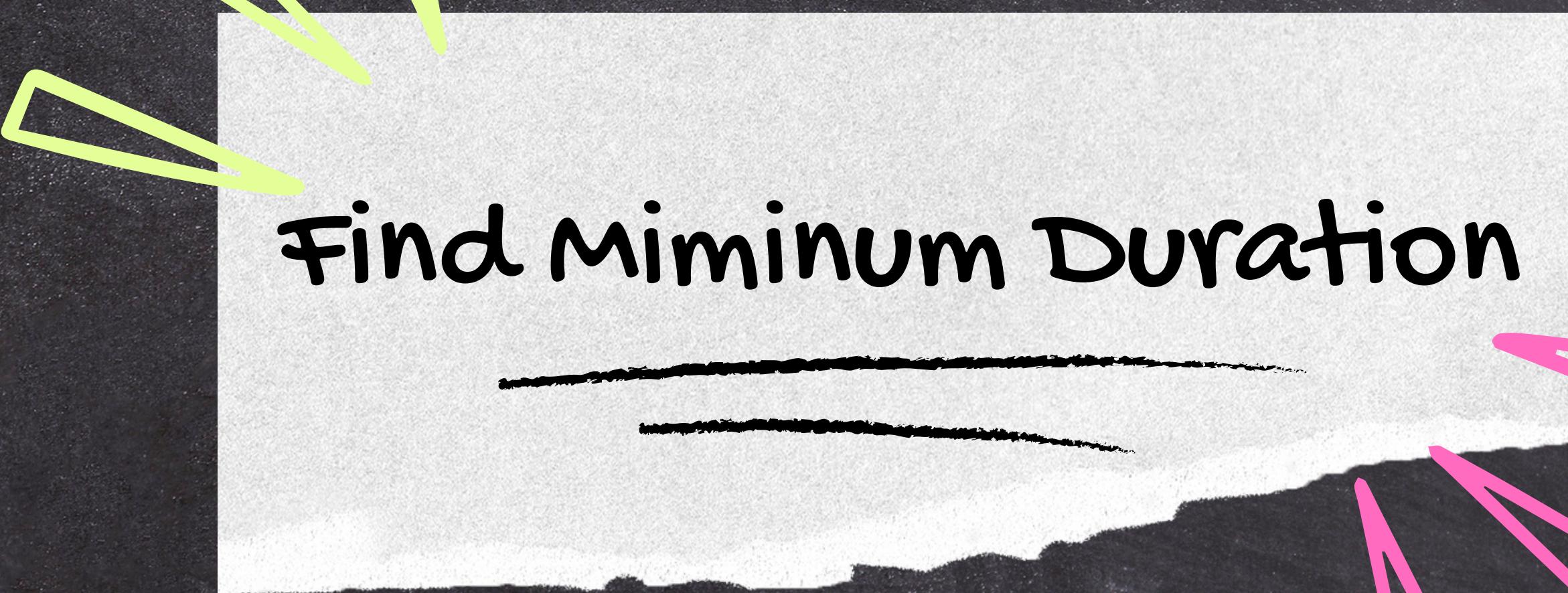
# Dark Matter

Use to increase the speed of the spaceship.  
(halves the duration of trip)

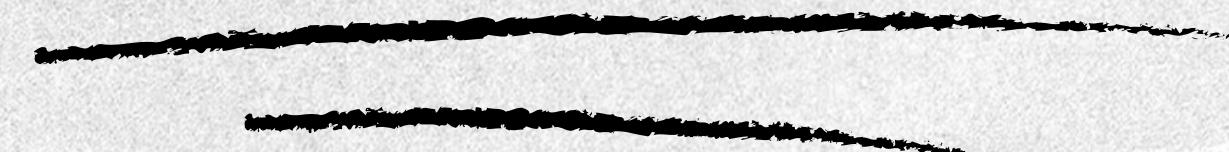
Only Use **one dark matter per trips** or  
spaceship goes boom

every trip, Nibbler expel a certain number of  
dark matter or none





Find Minimum Duration



# Inputs

**Each** test case

Given **n trips** (if n is 0, then stop programs)

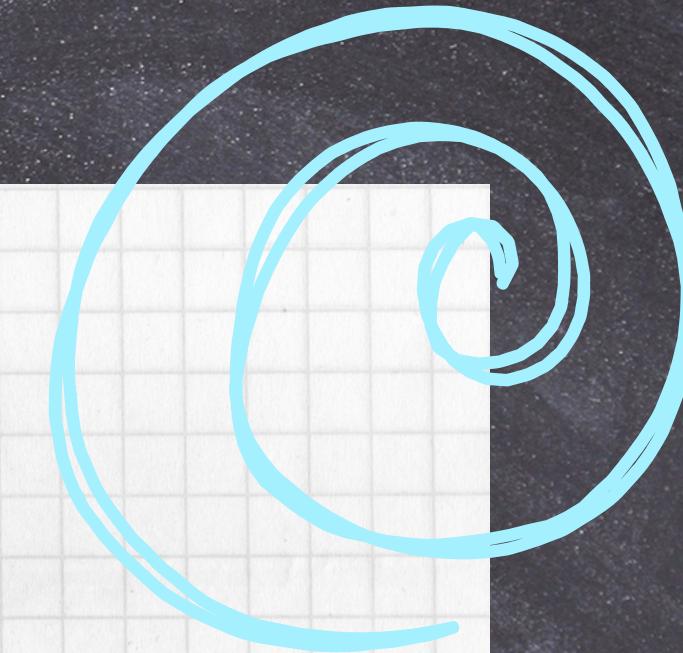
each trip get **duration of trip** & how many **dark matter** Nibbler  
expel.

Example:

2

24 1

10 0



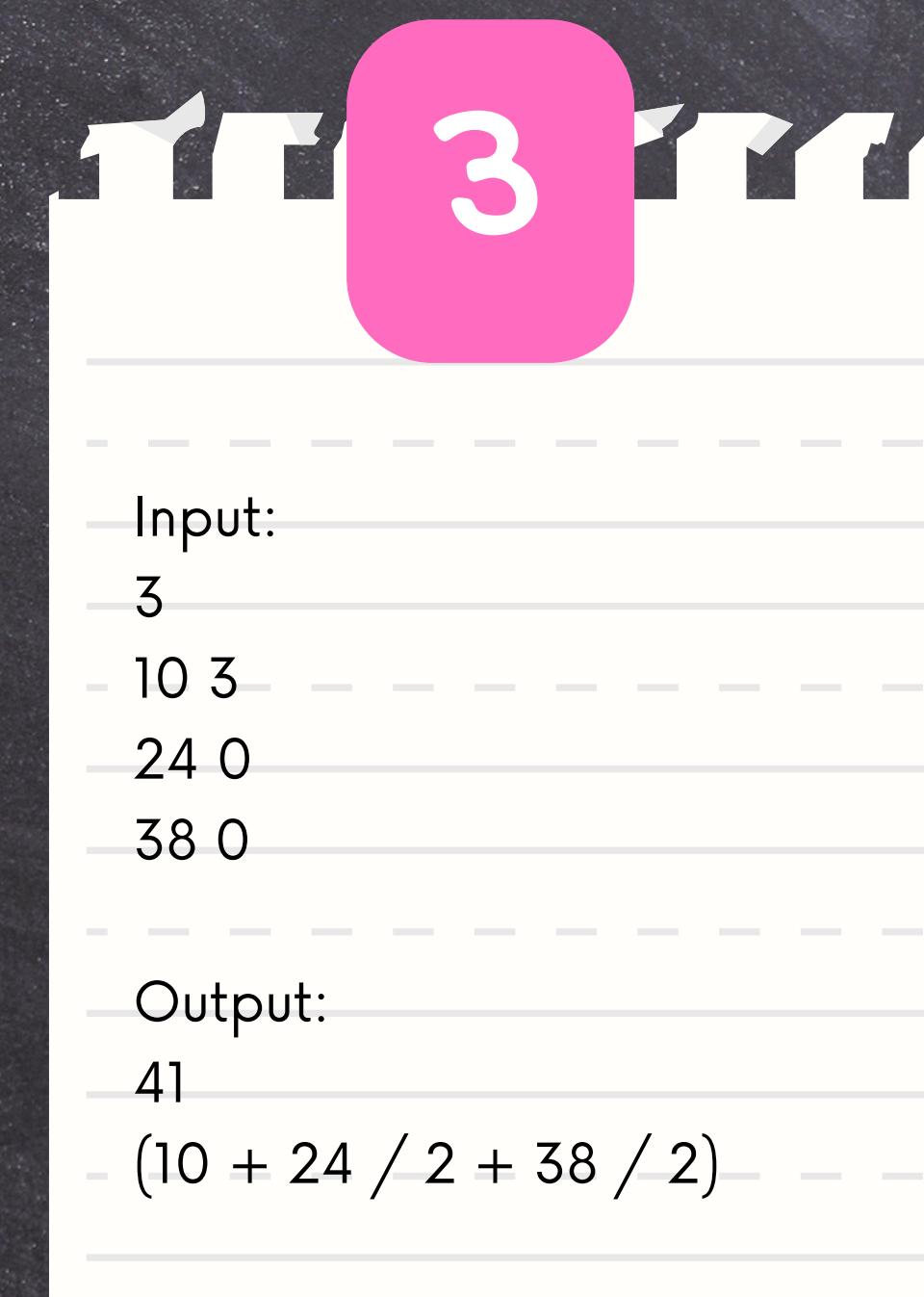
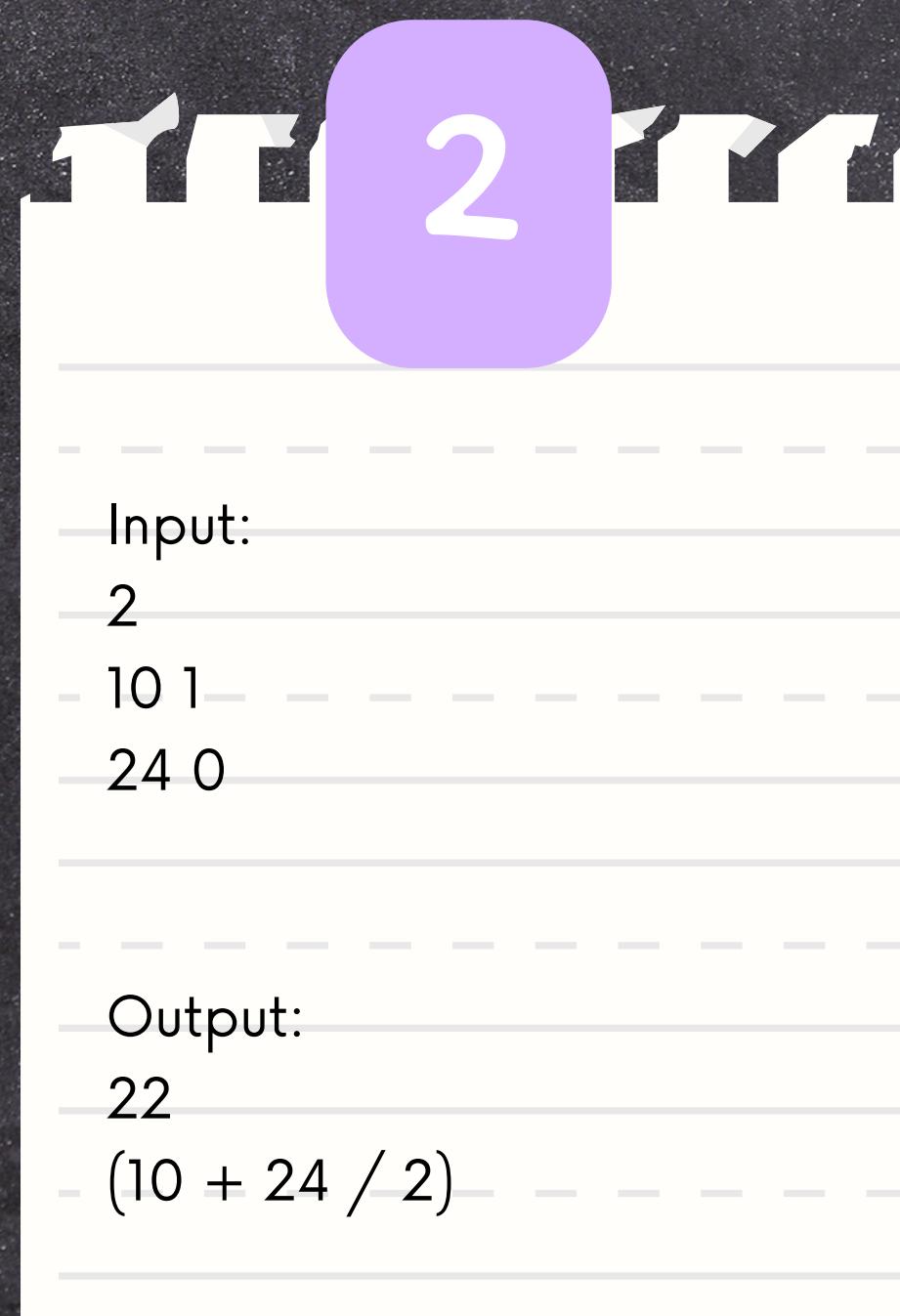
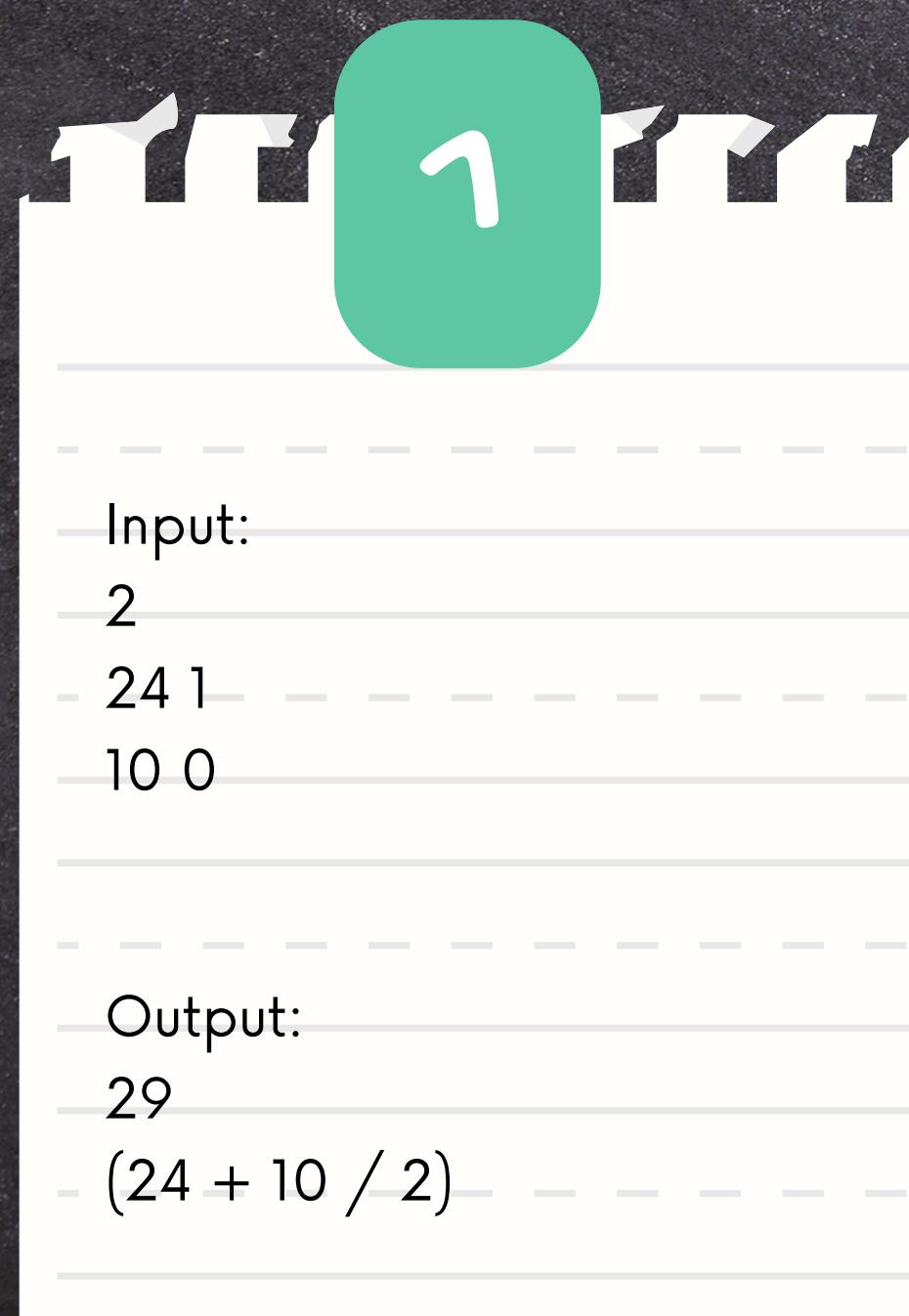
# Outputs

Every test case output the minimum duration of the trips.

Example:

29

# Example



# Limitation



## Trips

$$1 \leq n \leq 100$$

## DM

Dark Matter in  $T_i$ , can only be used in the upcoming trips only.

$$T_1, T_2, \dots, T_i, T_j, \dots, T_n$$

$$i < j \leq n$$

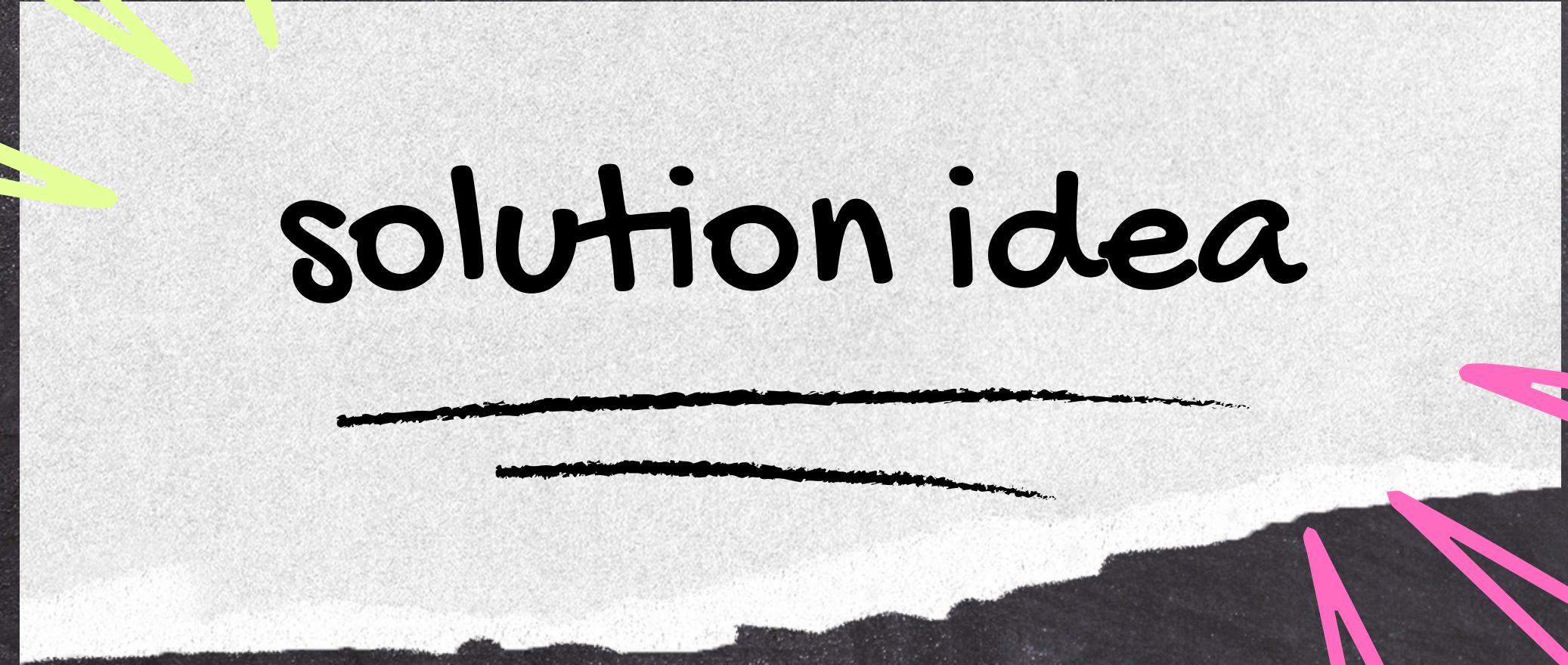
## Trip

Each trip given duration & dark matter ( $T_i, B_i$ )

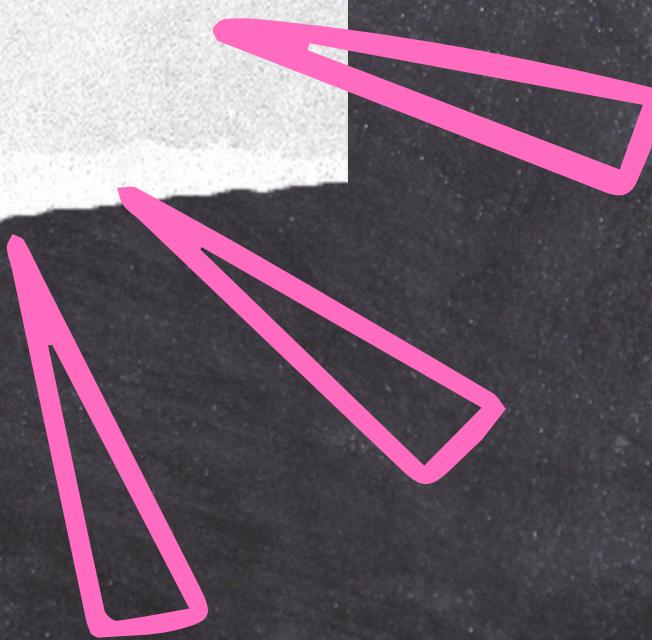
Duration:

- $2 \leq T_i \leq 1000$
- Always Even

Dark Matter:  $0 \leq B_i \leq n$



solution idea





Memoization

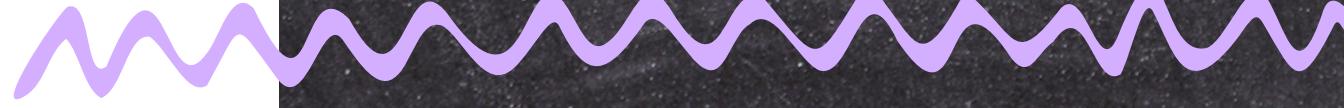
gramming matrix:

→ (sequence y)

1	2	3	4	5	6
T	G	C	T	C	G
-6	-12	-18	-24	-30	-36
5	-1	-7	-13	-19	-25
-1	3	-3	-2	-8	-14
-7	-3	8	2	3	-3
-13	-9	2	6	0	1
-19	-15	-4	7	4	-2
-25	-21	-10	1	5	2

gnment scores 11:

- - T C A T A  
G C T C G T A  
-6 -6 +5 +5 -2 +5 +5



# The X

Is The Number of trip on going

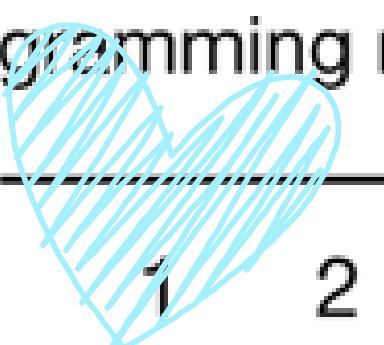


# The Y

Is The Number of Current Dark Matter



gramming matrix:

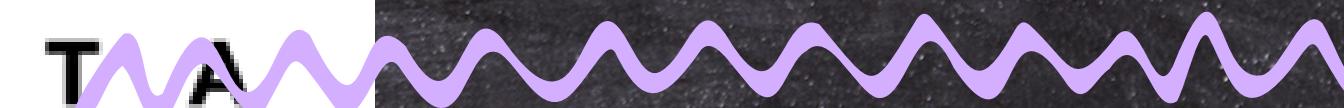


(sequence y)

1	2	3	4	5	6
T	G	C	T	C	G

-6	-12	-18	-24	-30	-36
5	-1	-7	-13	-19	-25
-1	3	-3	-2	-8	-14
-7	-3	8	2	3	-3
-13	-9	2	6	0	1
-19	-15	-4	7	4	-2
-25	-21	-10	1	5	2

gnment scores 11:

- - T C A T A  
G C T C G T   
-6 -6 +5 +5 -2 +5 +5



## solve(trip, dark matter)

Start at solve(0, 0)

Because the dark matter used in upcoming trips only

every recursive:

find minimum of distance on n trip

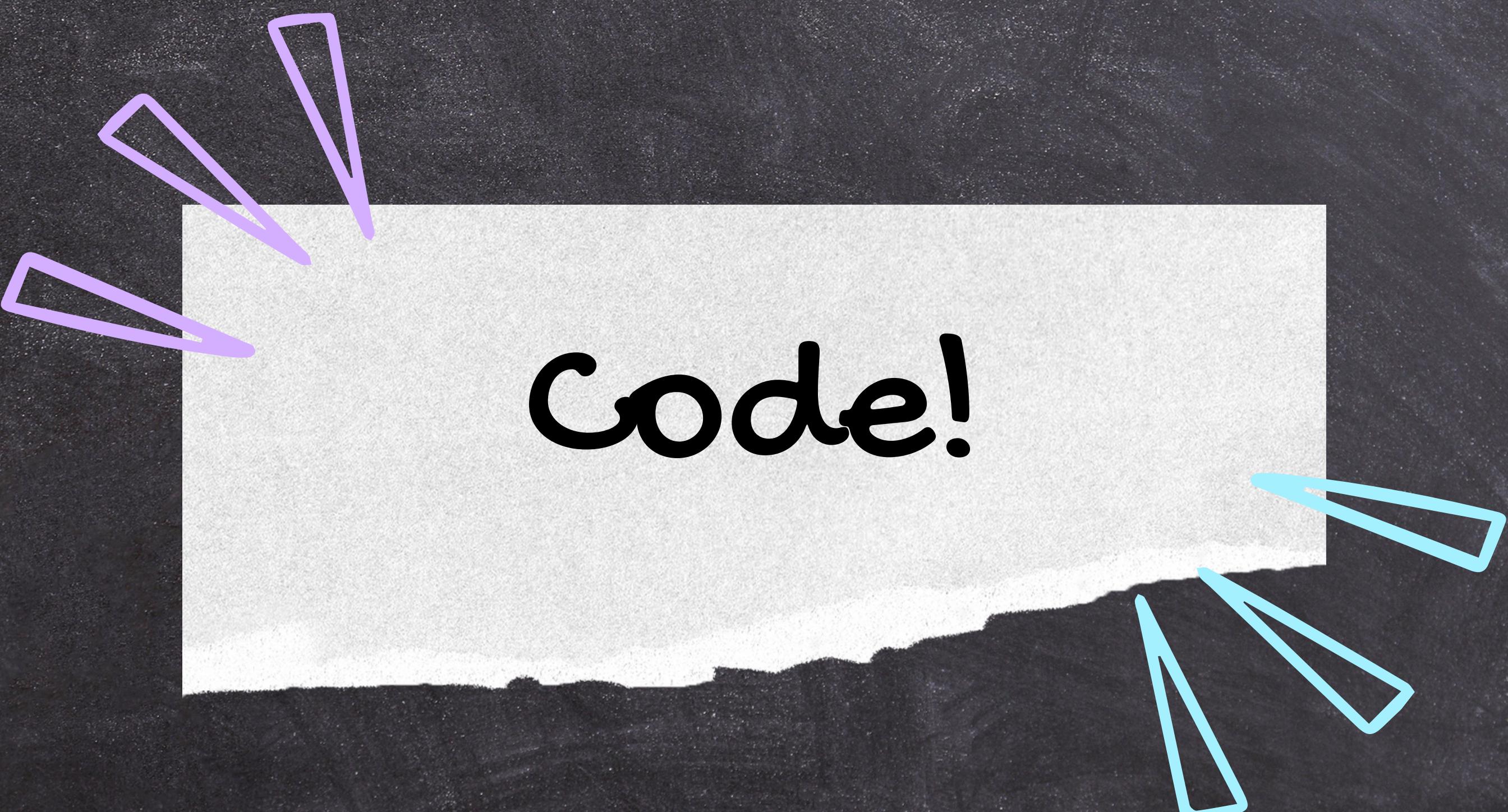
- with dark matter (if there's dark matter > 0):

duration[trip] / 2 + solve(trip + 1, dark matter - 1 + Nibbler[trip])

- without dark matter

duration[trip] + solve(trip + 1, dark matter + Nibbler[trip])





Code!

```
1 #define maxNum 2147483647;
2
3 int n;
4 pair<int, int> trips[110]; // distance, dark matter gen
5 int dp[110][10010]; // many trips, current dark matter
6
7 int solve(int nTrip, int currDM)
8 {
9     if (nTrip == n)
10         return 0;
11     if (dp[nTrip][currDM] != -1)
12         return dp[nTrip][currDM];
13
14     int res = maxNum;
15
16     // take dark matter
17     if (currDM > 0)
18         res = min(res, solve(nTrip + 1, currDM - 1 + trips[nTrip].second) + trips[nTrip].first / 2);
19
20     // not take dark matter
21     res = min(res, solve(nTrip + 1, currDM + trips[nTrip].second) + trips[nTrip].first);
22
23     return dp[nTrip][currDM] = res;
24 }
```



THANK  
YOU