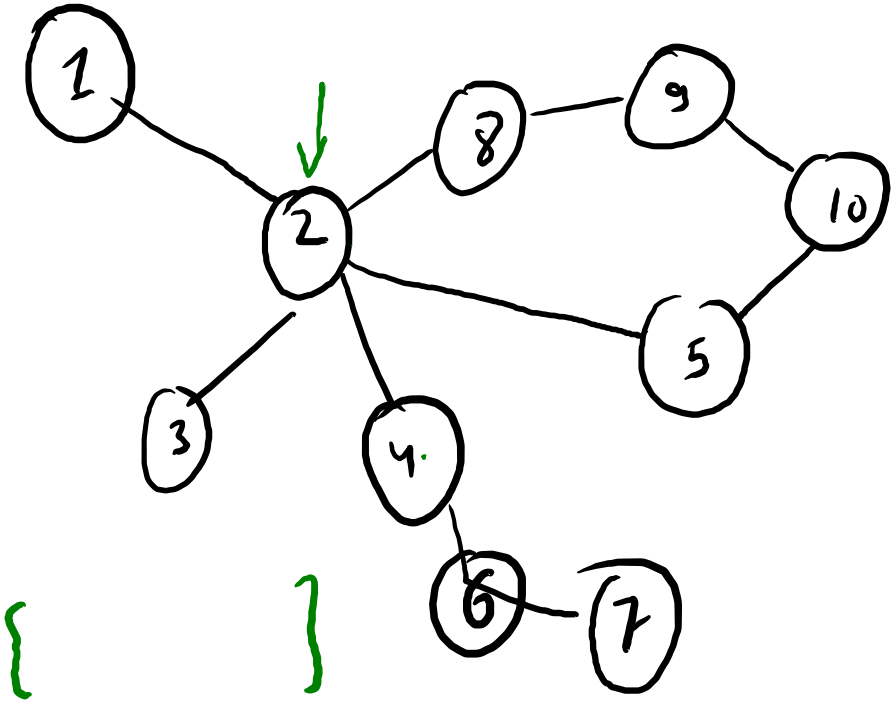
BFS

[2, 1, 3, 4, 5, 8, 6, 10, 9]

7

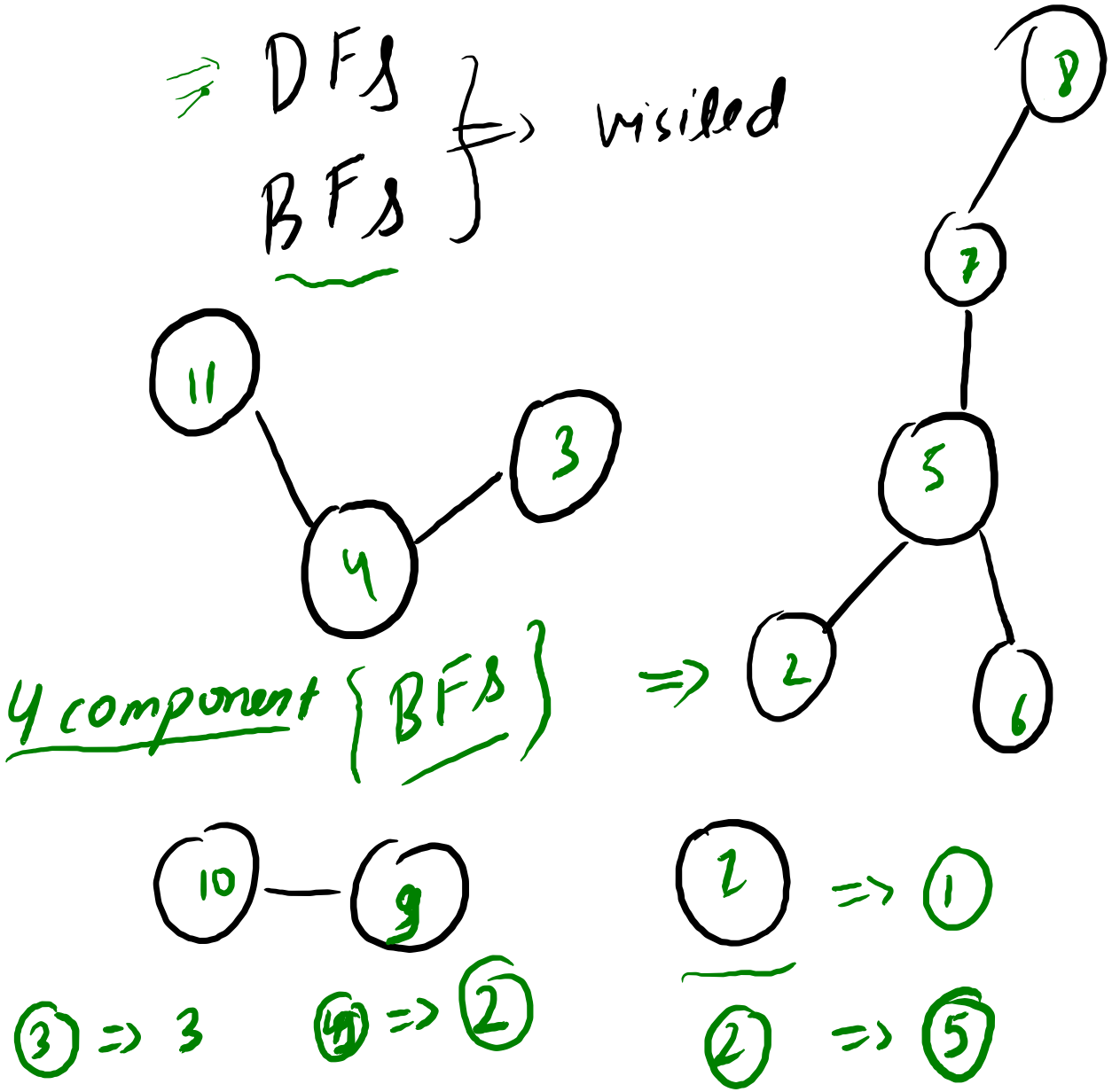
Queue

visited



2 → { ____ }

list


DFS
BFS
visited
11
3
4
8
7
5
2
6
4 component {BFS}
10
9
1
3
4
2
2
5

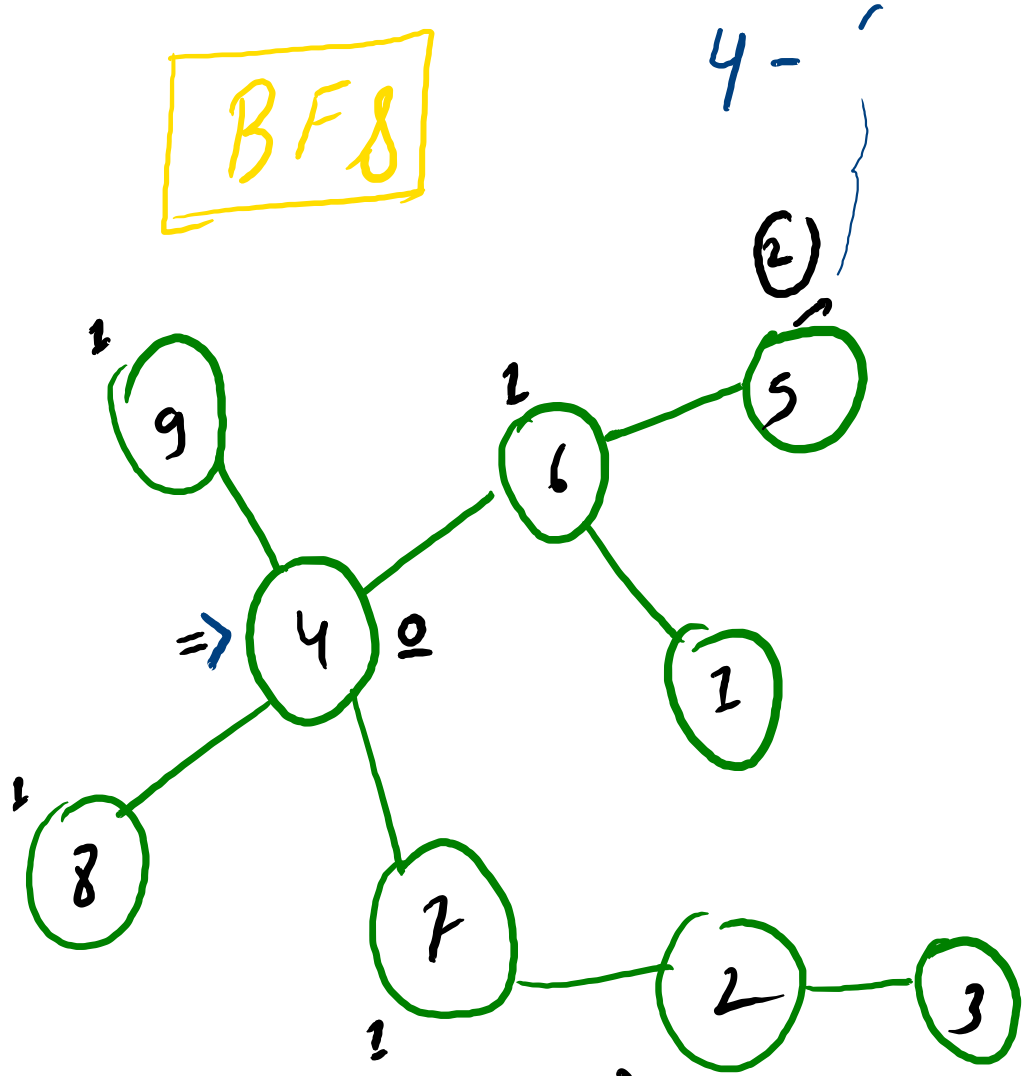
Staat => 4

[Input

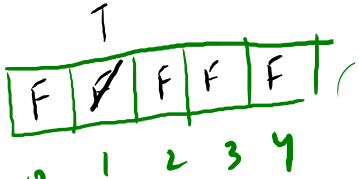
staat =>

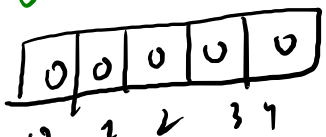
Q	
7	1
5	2
3	3
4	0

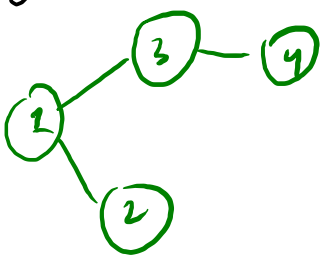
BFS



{ 5 min } => BFS







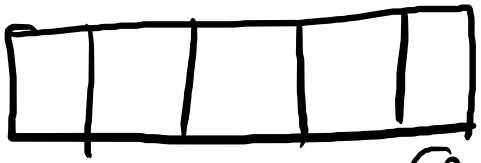

```
bfs( graph, start, distance[ ]) {
                visited[ ]
  LL<Integer> queue;
    queue.add( start);
    visited[start] = true;
    while( queue.size() != 0) {
      int x = queue.poll(); {First element}
      list = graph.get(x);
        for( int y : list ) {
          if(!visited[y]) {
            queue.add(y);
            visited[y]
          distance[y] = distance[x]+1;
        }
      }
}
```

distance

Q query

→ distance

1



0	1	2	3	4

BFS

Q

$N = 10^5$

$M = 10^5$

$Q = 10^5$

$(N + M + Q)$ 10^5

Q

BFS → N + M

Q

Y → distance[y]

BFS

10^{10} ⇒ ___

1 sec = 1 Clock speed

1 GHz $\Rightarrow 10^9$ ins

for (int i=0; i<10; i++)

1 it = 10 ins

1 GHz $= 10^9$ ins

2.4 GHz

1 sec = 2.4 GHz

1 sec = 2.4×10^9 ins

1 sec = 10^8 iteration

10^9 itr = 10^9 ins

10^{10} itr

(N+M)

N x Q

$10^5 \times 10^5$

10^{10} itr

a = 5 ± 10

$N = 10^5$

$N^2 \Rightarrow 10^{10}$

$N \log N$

$N = 10^7$

$N^2 = \boxed{10^9}$

1 sec $= 10^8$ itr

100 sec $= 10^{10}$ itr

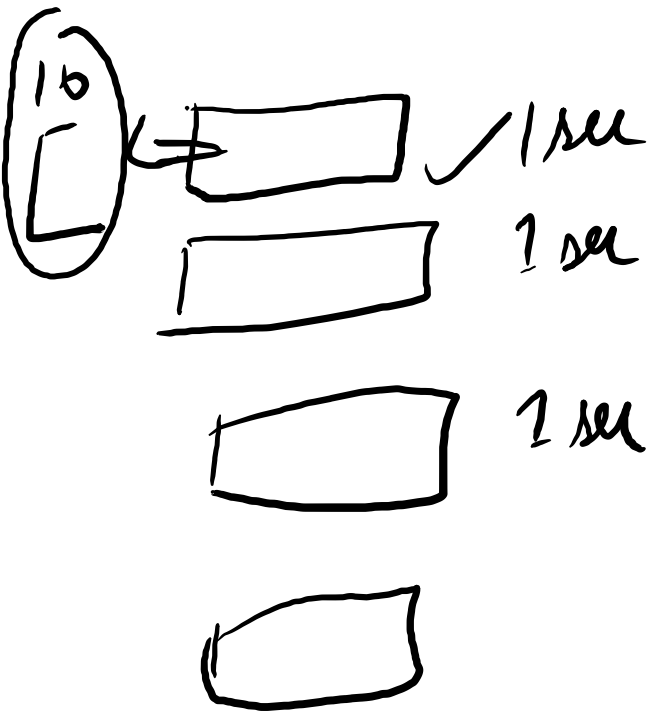
↓

1 min 40 sec

1 sec $= 10^8$ itr

⇓

TLE



$T^{N\log N}$

for ( ) {

BS

}

$T \times N$

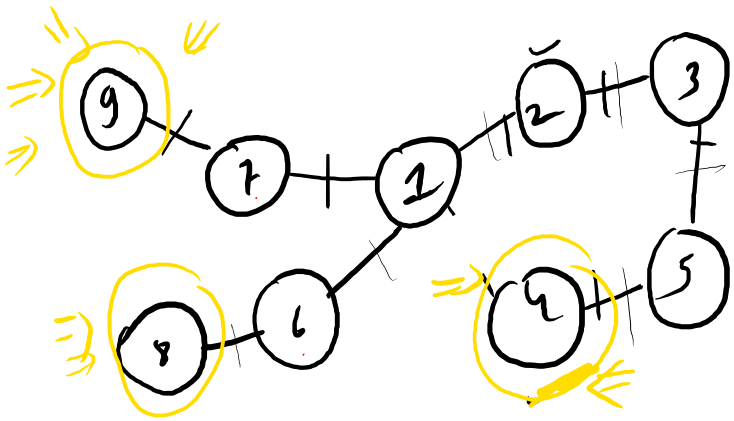
$10^4 \times 10^4 = 10^8$

10^4

$T \times 10^5$

$T \times \log N$

$N\log N + T \times \log N$



start
4

4

4	3	2	0	1	5	5	6	6
0	1	2	3	4	5	6	7	8	9

⇒ max 6 ⇒ 9

0	2	3	4	6	5	1	3	0	4
0	1	2	3	4					

4 → 9

~~2~~ ~~1~~ ~~3~~ ~~7~~ ~~6~~ ~~5~~ 9 8 4

1 → 4

9 → 4

8 → 4

BFS [start → any]
↳ last element in the queue.

BFS
↳ start = last
found the distance then max distance and it's index

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	0	1	2	3	1	1	1		1	2	2	1		2		0	2

7, 6, 2, 5, 17

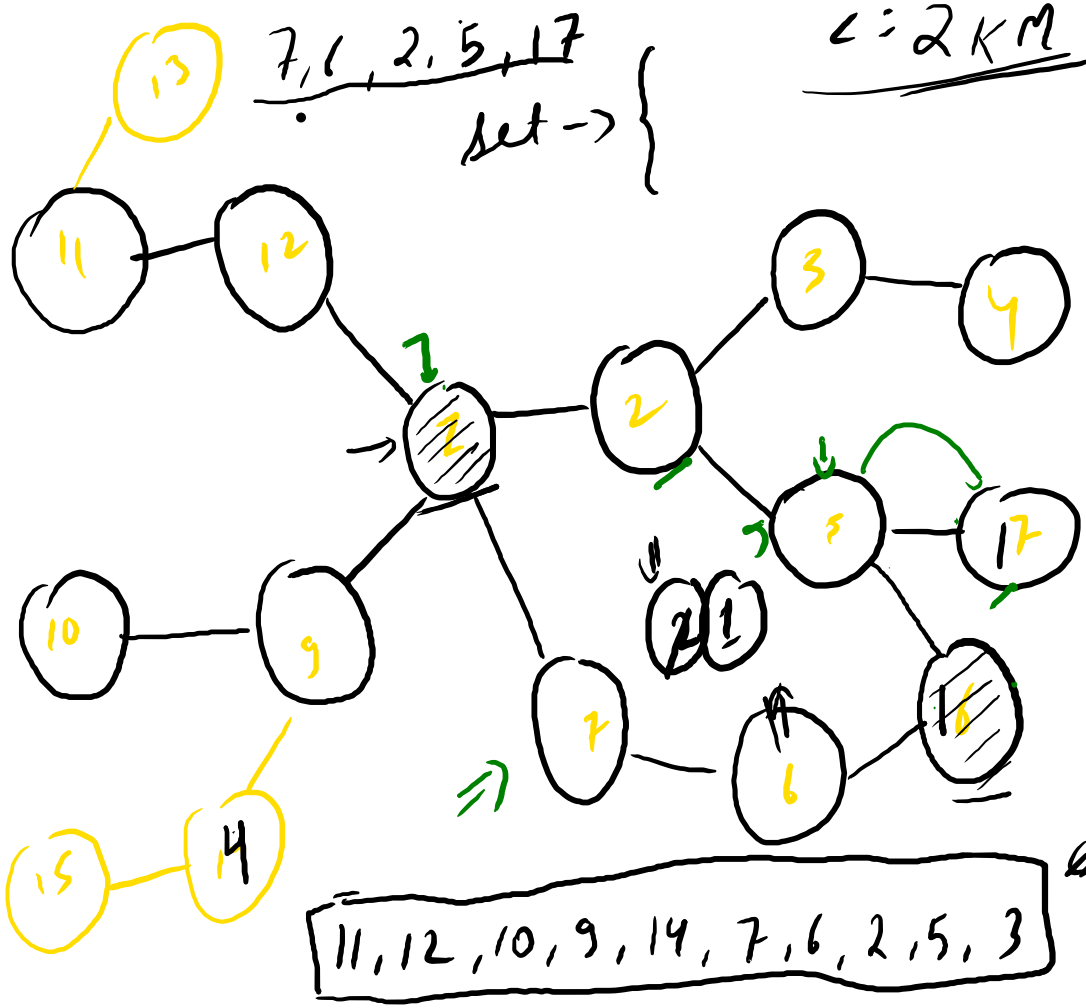
set -> {

c = 2KM

{~~1~~, ~~16~~} => ~~2~~ ~~7~~ ~~9~~ ~~12~~

~~5~~, ~~6~~, ~~3~~, 10, 14

11, 17



$\frac{N}{2} \times BFS$

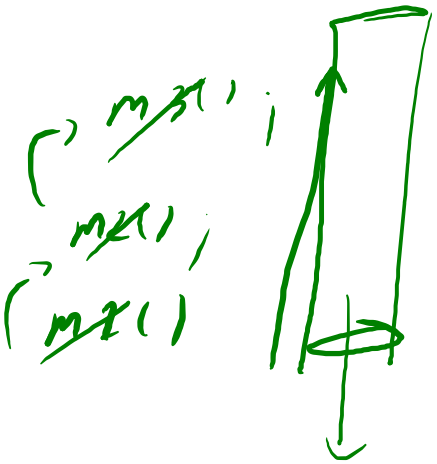
$\frac{N}{4} \times (N+M)$

$N^2 \Rightarrow TLE$

BFS

DISTANCE

11, 12, 10, 9, 14, 7, 6, 2, 5, 3

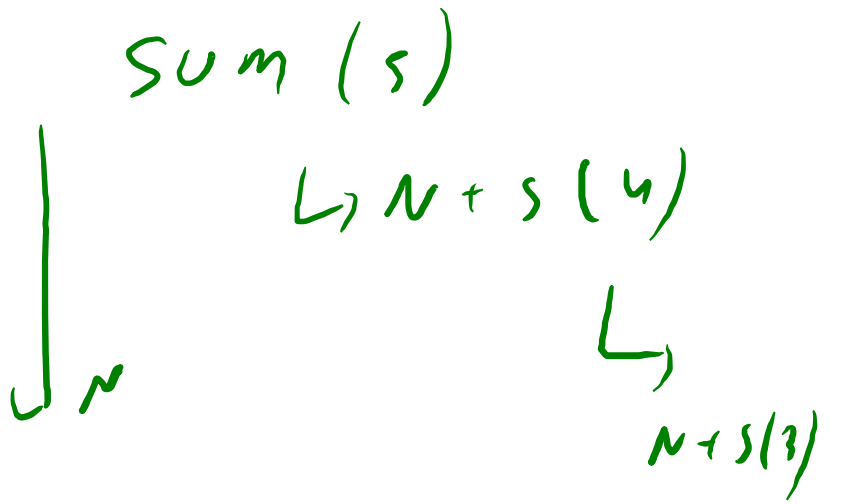


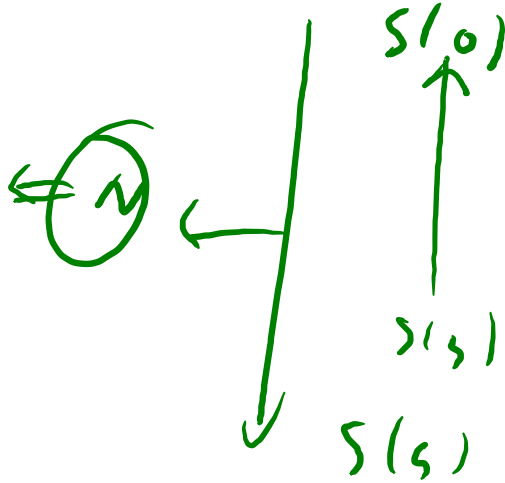
```
m2() {
    m2();
}
```

```
m2() {
    m3();
}
```

```
m3();
```

10


sum(s)
N + s(4)
N
N+s(3)


s(0)
N
s(3)
s(s)

1. Add all the domino's house into the queue
2. BFS update the distance
3. Iterate and count the index having distance less than or equal K