169. Majority Element

$$arr = \begin{bmatrix} \overset{0}{2}, & \overset{1}{2}, & \overset{2}{3}, & \overset{3}{2}, & \overset{4}{4}, & \overset{5}{4}, & \overset{6}{4}, & \overset{7}{4}, & \overset{8}{2}, & \overset{9}{2}, & \overset{10}{2}, & \overset{11}{2} \end{bmatrix}$$

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

dry
run

majority element = ~~2~~ ~~4~~ 2

count = ~~1~~ ~~2~~ ~~1~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ 2

psudo code

1) declare majority element = arr[0]

2) " count = 1

3) iterate from 1 to (n-1)

   3.1) if curr == majority ele

      count ++;

   3.2)

      count --;

      3.2.1) if count == 0 then

         majority element = arr[i];

         count = 1

   3.3) return majority element.

Code $T.C = O(n), \quad S.C = O(1)$

```java
public int majorityElement(int[] arr) {
    int majorityEle = arr[0];
    int count = 1;
    for (int i = 1; i < arr.length; i++) {
        if ( arr[i] == majorityEle ) {
            count++;
        } else {
            count--;
            if ( count == 0 ) {
                majorityEle = arr[i];
                count = 1;
            }
        }
    }
    return majorityEle;
}
```

$$arr = [2, 2, 2, 2, 1, 3, 4, 1, 3, 2, 2, 1, 2, 3, 2]$$

major ele = ~~2~~ ~~1~~ ~~3~~ 2

count = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~2~~ 2

Constructors 1_1

Code

```java
public class Solution {
    public static class Movie {
        String name;
        int rating;
        int money;
        int profit;
        String actor;
        String actress;

        public Movie(String name, int rating, int money, int profit, String actor, String actress) {
            this.name = name;
            this.rating = rating;
            this.money = money;
            this.profit = profit;
            this.actor = actor;
            this.actress = actress;
        }
    }
    public static void main(String[] args) {
        Movie batman1 = new Movie("Batman 1", 8, 200000, 5000, "John", "Disha");
        System.out.println(batman1.rating);
        System.out.println(batman1.money);
        System.out.println(batman1.profit);
        System.out.println(batman1.actor);
        System.out.println(batman1.actress);

        Movie batman2 = new Movie("Batman 2", 9, 500000, 8000, "Akshay", "Anushka");
        System.out.println(batman2.rating);
        System.out.println(batman2.money);
        System.out.println(batman2.profit);
        System.out.println(batman2.actor);
        System.out.println(batman2.actress);

        Movie batman3 = new Movie("Batman 3", 10, 700000, 6000, "Ajay", "Deepika");
        System.out.println(batman3.rating);
        System.out.println(batman3.money);
        System.out.println(batman3.profit);
        System.out.println(batman3.actor);
        System.out.println(batman3.actress);
    }
}
```
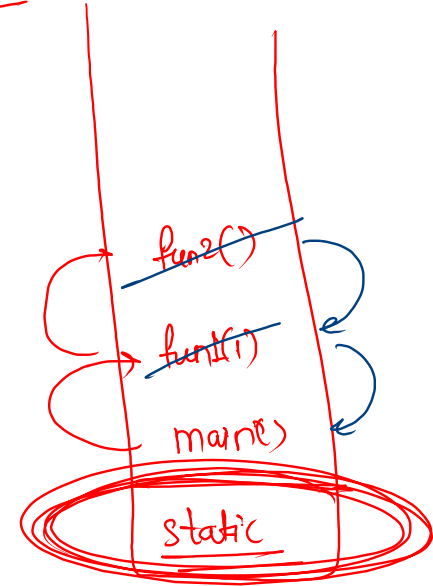
short

fun2()

fun1()

main()

static

Functions 1_3

```java
public static class Movie {
    // fields
    String name;
    int rating;
    int moneyCollection;
    int profit;
    String actor;
    String actress;

    public Movie(String name, int rating, int moneyCollection, int profit, String actor, String actress) {
        this.name = name;
        this.rating = rating;
        this.moneyCollection = moneyCollection;
        this.profit = profit;
        this.actor = actor;
        this.actress = actress;
    }

    public void propDisplay() {
        System.out.println(name);
        System.out.println(rating);
        System.out.println(moneyCollection);
        System.out.println(profit);
        System.out.println(actor);
        System.out.println(actress);
    }
}

public static void main(String[] args) {
    Movie batman1 = new Movie("Batman 1", 8, 200000, 5000, "Rachel Gupta & Nikhil Chinapa", "Disha");
    batman1.propDisplay();

    Movie batman2 = new Movie("Batman 2", 9, 500000, 8000, "Rannvijay Singha & Prince Narula", "Neha Dhupia");
    batman2.propDisplay();

    Movie batman3 = new Movie("Batman 3", 10, 700000, 6000, "Rachel Gupta & Prince Narula", "Neha Dhupia");
    batman3.propDisplay();
}
```

Methods 1_2

```java
public static class Movie {
    // fields
    String name;
    int rating;
    int moneyCollection;
    int profit;
    String actor;
    String actress;

    public Movie(String name, int rating, int moneyCollection, int profit, String actor, String actress) {
        this.name = name;
        this.rating = rating;
        this.moneyCollection = moneyCollection;
        this.profit = profit;
        this.actor = actor;
        this.actress = actress;
    }

    public void propDisplay() {
        System.out.println(name);
        System.out.println(rating);
        System.out.println(moneyCollection);
        System.out.println(profit);
        System.out.println(actor);
        System.out.println(actress);
    }

    public boolean checkHit() {
        return rating > 5;
    }
}

public static void main(String[] args) {
    Movie batman1 = new Movie("Batman 1", 8, 200000, 5000, "Rachel Gupta & Nikhil Chinapa", "Disha");
    System.out.println(batman1.checkHit());

    Movie batman2 = new Movie("Batman 2", 9, 500000, 8000, "Rannvijay Singha & Prince Narula", "Neha Dhupia");
    System.out.println(batman2.checkHit());

    Movie batman3 = new Movie("Batman 3", 4, 700000, 6000, "Rachel Gupta & Prince Narula", "Neha Dhupia");
    System.out.println(batman3.checkHit());
}
```

Constructor Overloading 1_1

```java
public static class Movie {
    // fields
    String name;
    int rating;
    int moneyCollection;
    int profit;
    String actor;
    String actress;

    public Movie(int rating, int moneyCollection) {
        this.rating = rating;
        this.moneyCollection = moneyCollection;
        System.out.println("Stored rating and money.");
    }

    public Movie(int rating, int moneyCollection, int profit) {
        this.rating = rating;
        this.moneyCollection = moneyCollection;
        this.profit = profit;
        System.out.println("Stored rating, money, profit are given.");
    }

    public Movie(int rating, int moneyCollection, int profit, String actor) {
        this.rating = rating;
        this.moneyCollection = moneyCollection;
        this.profit = profit;
        this.actor = actor;
        System.out.println("Stored rating, money, profit, lead actors are given.");
    }
}

public static void main(String[] args) {
    Movie superman1 = new Movie(8, 900000);
    Movie superman2 = new Movie(8, 900000, 8000);
    Movie superman3 = new Movie(8, 900000, 8000, "Ranveer Singh");
}
```

Note:- there is a |static block| (Imp)

     ↳ Imp:- This will always be the first thing
        to run when main f^ is called

Syntex:-

```
static {
        // statement

}
```

```java
class Movie {
    // fields
    String name;
    int rating;
    int moneyCollection;
    int profit;
    String actor;
    String actress;
    static String videoType;

    static {
        videoType = "movies";
    }

    public Movie(String name, int rating, int moneyCollection, int profit, String actor, String actress) {
        this.name = name;
        this.rating = rating;
        this.moneyCollection = moneyCollection;
        this.profit = profit;
        this.actor = actor;
        this.actress = actress;
    }
}

public class Solution {

    public static void main(String[] args) {
        Movie superman1 = new Movie("Superman 1" , 8, 900000,8000, "Rachel Gupta","Vaani");

        System.out.println(superman1.videoType);
        System.out.println(Movie.videoType);
    }
}
```

Note:-   final

const ⟶ JS

↑
Keyword

Leetcode

- → 735
- → 42
- → 739
- → 128 → Hint — DP