



گزارش پروژه میان ترم طراحی الگوریتم ها

نام دانشجو : علیرضا سعیدنیا

شماره دانشجویی : ۴۰۰۱۰۸۳۳

استاد:

دکتر محمد هادی علائیان

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فهرست مطالب

صفحه	عنوان
۴.....	تشریح الگوریتم
۵.....	مثال الگوریتم
۶.....	تشریح کد

تشریح الگوریتم

برای حل این مساله از یک آرایه استفاده می کنیم که متشکل از ۰ و ۱ ها است. ۰ به نشانه دروغگو بودن و ۱ به نشانه راستگو بودن. هدف مساله ، پیدا کردن افراد راستگو در جامعه ای متشکل از راستگوها و دروغگو ها است. برای حل این مساله به روش تقسیم و حل و بازگشتی ، آرایه را به ۲ قسمت تقسیم می کنیم. برای نصف کردن آرایه باید یک عنصر به نام عنصر mid پیدا کنیم که برابر با میانگین ایندکس چپ و راست آرایه ما می باشد. یکبار برای نصفه چپ اینکار را می کنیم و یک بار برای نصفه راست ، دوباره برای نصفه نصفه چپ و و غیره. اینکار را تا ۲ تایی شدن اعضا ادامه می دهیم. اگر اعضای آرایه زوج و توانی از ۲ باشد ، تعداد زیرمجموعه های دوتایی $\frac{n}{2}$ در می آید. در غیر این صورت به تعداد ۳ تایی هم در می آید. مثلاً یک آرایه ۷ عضوی ، به دوتا دو عضوی و یک سه عضوی تقسیم می شود.

چون برای حل مساله ، نیاز به مقایسه جفت شوالیه ها داریم ، در صورتی که تعداد اعضای آرایه دو عضوی باشد ، برای هر زیر مجموعه دو عضوی این مقایسه را انجام می دهیم. اگر تعداد به صورتی باشد که ۳ عضوی داشته باشیم ، یکی از اعضای راست یا چپ زیرمجموعه سه عضوی را انتخاب می کنیم (در اینجا من راست را انتخاب می کنم) و با دورترین عدد زیر مجموعه ۳ عضوی (در اینجا چپ ترین عضو) مقایسه می کنیم و در صورت نیاز ایندکس آن را چاپ کرده و سپس آن را دور می ریزیم (منظور از دور ریختن ، این است که یکبار دیگه تابع بازگشتی ام را صدا می زنم و آن عدد دیگر شامل ایندکس end ما نمی باشد) اگر راستگو بود ایندکس آن برگردانده می شود. اگر دروغگو بود ، چون ما در این مساله دنبال راستگو هستیم ، آن را دور می ریزیم. مثلاً فرض کنید می دانیم سمت راستی راستگو و دورترین دروغگو باشد ، راستگو به دروغگو می گوید دروغگو ، ایندکس راستگو برگردانده می شود و سپس دور ریخته می شود.

فرض کنید می دانیم سمت راستی راست می گوید و چپ ترین راست می گوید ، جفتشان به هم می گویند راستگو ، ایندکس سمت راستی چاپ و عدد سمت راست ، دور ریخته می شود. فرض کنید می دانیم جفتشان دروغگو باشند ، چون دنبال پیدا کردن راستگو هستیم و این دروغگو هرچه درمورد یکدیگر بگویند درست نیست و دروغ می باشد . باز هم سمت راستی حذف می شود. هدف ما از حذف کردن سمت راستی این بود که بیاییم و زیرمجموعه ۳ عضوی را تبدیل به یک زیرمجموعه دو عضوی کنیم.

اگر زیرمجموعه دو عضوی باشد :

- اگر جفت راست بگویند ، ایندکس هردو پرینت می شود.
- اگر یکی راست و یکی دروغ بگوید ، ایندکس راستگو پرینت میشود.
- اگر هردو دروغ بگویند ، ایندکس هیچکس پرینت نمی شود . چون راستگویی وجود ندارد.

بدین صورت با مرتبه زمانی $O(n)$ الگوریتم اجرا می شود.

مثال الگوریتم

$[0,0,1,1,1,0,1]$

$[0,0,1,1][1,0,1]$

$[0,0][1,1][1,0,1]$

چون زیر مجموعه ۳ عضوی داریم. قرمز (راست ترین) و آبی (چپ ترین) باید مقایسه شوند. چون قرمز و آبی دارند راست می گویند، ایندکس قرمز پرینت شده و دور ریخته می شود.

$[0,0][1,1][1,0]$

همه چیز دوتایی شد. حال مقایسه به راحتی انجام می گردد . در چپ ترین زیرمجموعه دو عضوی جفت دروغ می گویند پس هیچ کاری نمیکنیم. در زیرمجموعه دو عضوی وسط ، چون هردو راست می گویند ، ایندکس جفت

آن هارا چاپ می کنیم. در سمت راست هم عضو سمت چپ راستگو است و راستی دروغگو است. راستگو را چاپ می کنیم.

تشریح کد

چون تعداد راست گویان ، همیشه باید از نصف اعضای آرایه بیشتر باشد. پس حداقل سه عضو داریم . پس اگر اعضا کوچکتر از ۳ باشد مشکل داریم.

```
Scanner scanner = new Scanner(System.in);
int howMany = scanner.nextInt();
if(howMany<3){
    System.out.println("no result!");
    System.exit( status: 0);
}
```

```
int randomizer_honest=random.nextInt( origin: howMany/2+1, bound: howMany+1);
int [] arrayOfKnight=new int[howMany];
while(randomizer_honest!=0){
    int rand=random.nextInt(howMany);
    arrayOfKnight[rand]=1;
    randomizer_honest--;
}
```

در سوال مطرح شده است که تعداد اعضای نفرات راستگو باید از نصف بیشتر باشد. در خط اول عکس بالا تعداد اعضای راستگو را میسازیم که از نصف بیشتر است و حداکثر می تواند تا تعداد اعضا برود.

به تعداد اعضا یک آرایه می سازیم. اعضای آرایه قبل از مقدار دهی ۰ هستند. حال وقت آن است که اعضای راستگو را وارد آرایه کنیم (در خانه های رندوم) ، پس با استفاده از تابع رندوم یک عدد از ۰ تا تعداد اعضای آرایه منهای یک می گیریم که میشود ایندکس ما ، سپس در آن نقطه رندوم مقدار ۱ را قرار می دهیم.

```
recursion(arrayOfKnight, begin: 0, end: arrayOfKnight.length-1);
```

تابع بازگشتی مان را با آرایه کلی صدا می زنیم.

```
int mid=(end+begin)/2;
```

همانطور که ذکر شد ، برای اینکه عمل نصف شدن انجام شود ، باید عنصر وسط پیدا شود.

```
recursion(array, begin, mid);
recursion(array, begin: mid+1, end);
```

در بخش آخر تابع بازگشتی برای نصفه چپ و نصفه راست دوباره تابع را صدا می زنیم. خود تابع به دو بخش اینکه آیا زیرمجموعه ۲ عضوی است یا ۳ عضوی تقسیم می شود.

```
if(end-begin+1==2){
    if(array[begin]==1&&array[end]==1){
        System.out.print(begin+" "+end+" ");
        return;
    }
    else if(array[begin]==1&&array[end]==0){
        System.out.print(begin+" ");
        return;
    }
    else if(array[begin]==0&&array[end]==1){
        System.out.print(end+" ");
        return;
    }
    else if(array[begin]==0&&array[end]==0)
        return;
}
```

اگر آن زیرمجموعه اختلاف ایندکس ابتدایی و انتهایی اش ۲ تا بود یعنی به خواسته خود رسیدیم ، حال طبق این مقایسه که در توصیف الگوریتم اشاره شد مقایسه را انجام می دهیم .

- اگر جفت راست بگویند ، ایندکس هردو پرینت می شود (هم begin هم end)
- اگر یکی راست و یکی دروغ بگویند ، ایندکس راستگو پرینت میشود. (فقط end)
- اگر هردو دروغ بگویند ، ایندکس هیچکس پرینت نمی شود . چون راستگویی وجود ندارد (ریترن خالی).

```

else if(end-begin+1==3){

    if(array[end]==0&&array[begin]==1|| array[end]==0&&array[begin]==0)
        recursion(array, begin, end: end-1);
    else if(array[end]==1&&array[begin]==0){
        recursion(array, begin: begin+1, end);
    }
    else if(array[end]==1&&array[begin]==1){
        System.out.print(end+" ");
        recursion(array, begin, end: end-1);
    }
    return;
}

```

اگر آن زیرمجموعه اختلاف ایندکس ابتدایی و انتهایی اش ۳ تا بود ، یعنی باید یکی را بریزیم دور.

- اگر سمت راستی دروغگو بود و سمت چپی (دورترین) راست گو بود، دروغگو را دور می ریزیم.
- اگر سمت راستی و چپی دروغگو بودند ، سمت راستی را دور می ریزیم.
- اگر سمت چپی دروغ بگوید و سمت راستی راست بگوید ، سمت راست را ایندکسش را پرینت می کنیم و آن را دور می ریزیم.
- اگر هم سمت چپی و هم سمت راستی راست بگویند ، ایندکس سمت راستی را چاپ می کنیم و آن را دور می ریزیم.

در آخر کاری که انجام داده ایم را ریترن می کنیم.