

دانشکده مهندسی کامپیوتر

**گزارش پروژه میان ترم طراحی الگوریتم ها**

**نام دانشجو : علیرضا سعیدنیا**

**شماره دانشجویی : 40010833**

**استاد:**

**دکتر محمد هادی علائیان**



**فهرست مطالب**

**عنوان صفحه**

|  |
| --- |
| تشریح الگوریتم ................................................................................................................................4  مثال الگوریتم ..................................................................................................................................5  تشریح کد .........................................................................................................................................6 |

**تشریح الگوریتم**

****

برای حل این مساله از یک آرایه استفاده می کنم که متشکل از 0 و 1 ها است. 0 به نشانه دروغگو بودن و 1 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 ، ارایه را به 2 قسمت تقسیم می کنیم. برای نصف کردن ارایه 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 2 تایی شدن اعضا ادامه می دهیم. اگر اعضای آرایه زوج و توانی از 2 باشد ، تعداد زیرمجموعه های دوتایی در می آید. در غیر این صورت به تعداد 3 تایی هم در می آید. مثلا یک ارایه 7 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 3 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 3 عضوی (در اینجا چپ ترین عضو) مقایسه می کنیم و در صورت نیاز ایندکس آن را چاپ کرده و سپس ان 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 ، ان را دور می ریزیم. مثلا 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 پیداکردن 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 3 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 الگوریتم اجرا می شود.

**مثال الگوریتم**

****

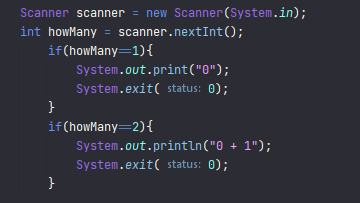
چون زیر مجموعه 3 عضوی داریم. قرمز (راست ترین) و آبی (چپ ترین) باید مقایسه شوند. چون قرمز و آبی دارند راست می گویند، ایندکس قرمز پرینت شده و دور ریخته می شود.

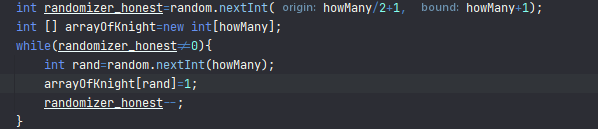
همه چیز دوتایی شد. حال مقایسه به راحتی انجام می گردد . در چپ ترین زیرمجموعه دو عضوی جفت دروغ می گویند پس هیچ کاری نمیکنیم. در زیرمجموعه دو عضوی وسط ، چون هردو راست می گویند ، ایندکس جفت آن هارا چاپ می کنیم. در سمت راست هم عضو سمت چپ راستگو است و راستی دروغگو است. راستگو را چاپ می کنیم.

**تشریح کد**

****

*اگر کلا یک یا دو شوالیه داشتیم ناچارا باید همین یا یا دو شوالیه راستگو باشند.*

**

****

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 0 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 0 تا تعداد اعضای ارایه منهای یک می گیریم که میشود ایندکس ما ، سپس در ان نقطه رندوم مقدار 1 را قرار می دهیم.



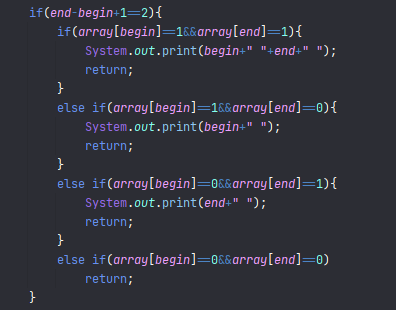
تابع بازگشتی مان را با آرایه کلی صدا می زنیم.



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

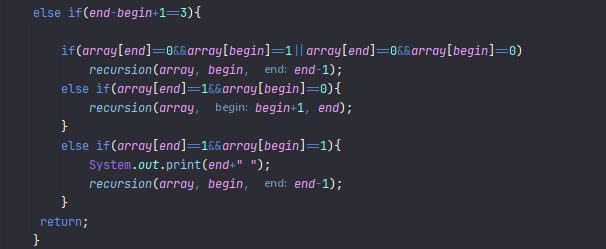


در بخش آخر تابع بازگشتی برای نصفه چپ و نصفه راست دوباره تابع را صدا می زنیم.خود تابع به دو بخش اینکه آیا زیرمجموعه 2 عضوی است یا 3 عضوی تقسیم می شود.



اگر آن زیرمجموعه اختلاف ایندکس ابتدایی و انتهایی اش 2 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 بگوید ، ایندکس راستگو پرینت میشود. ( فقط end )
* اگر هردو دروغ بگویند ، ایندکس هیچکس پرینت نمی شود . چون راستگویی وجود ندارد ( ریترن خالی).



اگر آن زیرمجموعه اختلاف ایندکس ابتدایی و انتهایی اش 3 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.