## پروژه درس ریاضیات گسسته <sub>D7</sub>

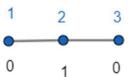
## على كريمي - سينا اخوى

9923503----9929733

## شرح حل سوال:

در این سوال ما بدنبال شماره گزاری راس های یک گراف هستیم و چون زوج یا فرد بودن راس برای ما مهم است در حل مسئله راس با تگ زوج را با عدد ۰ و راس با تگ فرد را با عدد ۱ مشخص می کنیم برای مثال در گراف روبرو که ۳ راس به هم پیوسته دارد داس که در در اس کراف روبرو که ۳ راس به هم پیوسته دارد در در اس کرانی شماده گذاری شده اند به این معند که

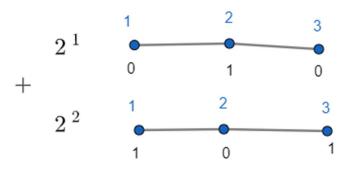
راس ۱و۳ با تَگ • و راس ۲ با تگ ۱ شماره گزاری شده اند به این معنی که راس ۱و۳ هرکدام میتوانند ۲ باشند و راس ۲ یا میتواند ۱ باشد یا ۳ چون از بین اعداد ۱۲۳ عدد ۲ زوج که تگ • و عدد ۱و۳ فرد که تگ ۱ که نمایانگر عدد فرد است.



$$(2^2 + 2^1) = 6$$

دو راس مجاور نمیتوانند هر دو تگ ۰ یا ۱ داشته باشند

همچنین بعد از اینکه به راس ها تگ و ا بدهیم میتوانیم جای و ها و ۱ ها را با هم عوض کنیم که هر ۱ دو حالت دارد بنابر این 2² حالت برای 1-0-1 و 2¹ حالت برای 0-1-0 داریم که در مجموع 6 حالت میشود در کد برنامه کا با java SE 11 نوشته شده است برای نمایش و تحلیل گراف و اطلاعات آن از جند کلاس استفاده شده است



```
public class Edge {
    private Vertex v1;
    private Vertex v2;
```

برای نمایش یال که از دو راس تشکیل شده است

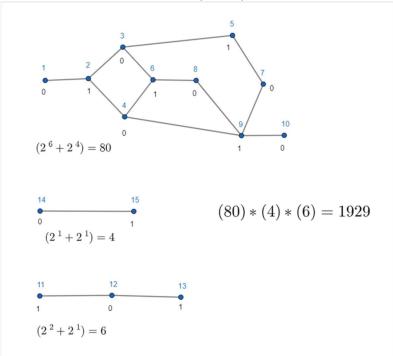
```
public class Vertex {
   private int index;
   private ArrayList<Vertex> adjacentVertices;
   private int tag = 3;
```

برای نمایش راس که شماره و یک لیست از راس ها مجاور و یک tag که تگ ۳ نشان دهنده این است که هنوز کامپوننت بندی (Component)(دسته بندی) نشده که تبدیل به چند گراف همبند شود تگ ۲ در برنامه نشان دهنده این است که کامپوننت بندی انجام شده و تگ ۰ و ۱ هم به ترتیب نشان دهنده زوج یا فرد بودن عدد انتساب شده به راس است

```
public class Graph {
    private GraphInfo graphInfo;
    private ArrayList<Vertex> vertices;
    private ArrayList<Edge> edges;
    private ArrayList<Component> components;
```

کلاس گراف که شامل اطلاعات گراف لیستی از راس ها و یال ها و لیستی از کامپوننت ها است برای مثال گراف زیر از سه کامپوننت تشکیل شده که هر کامپوننت چون مستقل از یکدیگر هستند به

طور مجزا تحلیل میشود و نتیجه نهایی در هم ضرب میشود و باقیمانده بر 998244353 نمایش داده میشود



## قسمتی از برنامه که برای تخصیص دادن تگ به راس هاست به شکل بازگشتی نوشته شده

```
public void setTagToVertexAndToComponentRecursively( Vertex
vertex, int assigningTag ) throws NotBeautifulGraphException {
   int oppositeTag = (assigningTag == 0 ? 1 : 0);
   int neutralTag = 2;

   if (vertex.getTag() == oppositeTag) {
        throw new NotBeautifulGraphException();
   }
   if (vertex.getTag() == assigningTag) {
        return;
   }
   if (vertex.getTag() == neutralTag) {
        vertex.setTag(assigningTag);
        for (Vertex adjacentVertex :
   vertex.getAdjacentVertices()) {
   setTagToVertexAndToComponentRecursively(adjacentVertex,
   oppositeTag);
```

```
}
    return;
}
throw new RuntimeException("should not reach this
statement");
}
```

همچنین برای ذخیره اطلاعات ورودی توسط کاربر و بعد تبدیل آن به گراف از کلاس GraphInfo استفاده شده ک

```
public class GraphInfo {
   private int noVertices;
   private int noEdges;
   private ArrayList<TwoNumber> edgesList;
```

کار گرفتن ورودی و نمایش خروجی به کاربر هم از طریق کلاس مجزای دیگری به نام console انجام شده

برای ذخیره سازی گراف از روش ماتریس ها استفاده نشده بلکه بر اساس لیست راس ها مجاور هر راس انجام شده تا تحلیل آسان تر و دست باز تری به ما برنامه نویسان بدهد