**ندا جمالی**

**9631813**

**طراحی الگوریتم**

**(گزارش پروژه)**

در این پروژه ابتدا یک کلاس نود می سازیم که نشان دهنده ی هر نود در گرافمان است. تنها ویژگی های این کلاس ذخیره همسایه ها و شماره نود مورد نظر است.

این کلاس همه ی نود ها را نکه می دارد و اگر بخواهیم که دوباره کلاسی که قبلا ساخته را دوباره بسازد کلاس جدیدی ساخته نمی شود بلکه کلاس قبلی برگشت داده می شود.همچنین در صورت نیاز به ایجاد ارتباط دو گرهلازم است که هر دو گره در همسایه های هر دو گره اضافه شود.

برای افزایش سرعت استفاده از کلاس ها از تابع هش استفاده شده است.

نمونه های کلاس نیز پیاده سازی شده اندکه می توانیم نمونه را داخل مجموعه نگه داریم.

در انتها تابع پیدا کردن مسیر بین دو نقطه به صورت DFSپیاده سازی شده است.

برای بخش امتیازی نیز از:

networkx

matplotlib

برای رسم گراف استفاده شده است.

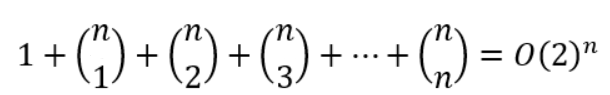
در ابتدا خود گراف نمایش داده می شود و سپس یکی یکی راه های پیدا شده به صورت مجزا نمایش داده می شوند.(با یال های قرمز)

**(پیچیدگی زمانی)**

از آنجایی که گراف کامل است می توانیم برای رفتن به یک نود از هر نودی شروع کنیم.

اگر بین نود مبدا و مقصد نودی نباشد 1 و اگر 1 نود دیگر باشد انتخاب 1 از n اگر 2 نود وجود داشته باشد انتخاب 2 از n و اگر n نود باشد انتخاب n ازn می باشد که مجموع را می توان 2 به توان n در نظر گرفت .

یعنی پیچیدگی زمانی برابر:



**(اجرای برنامه)**

**ورودی:**

**6**

**1 2**

**1 3**

**3 4**

**3 5**

**4 6**

**5 6**

**2 3**

**2 4**

**0 0**

**خروجی:**

