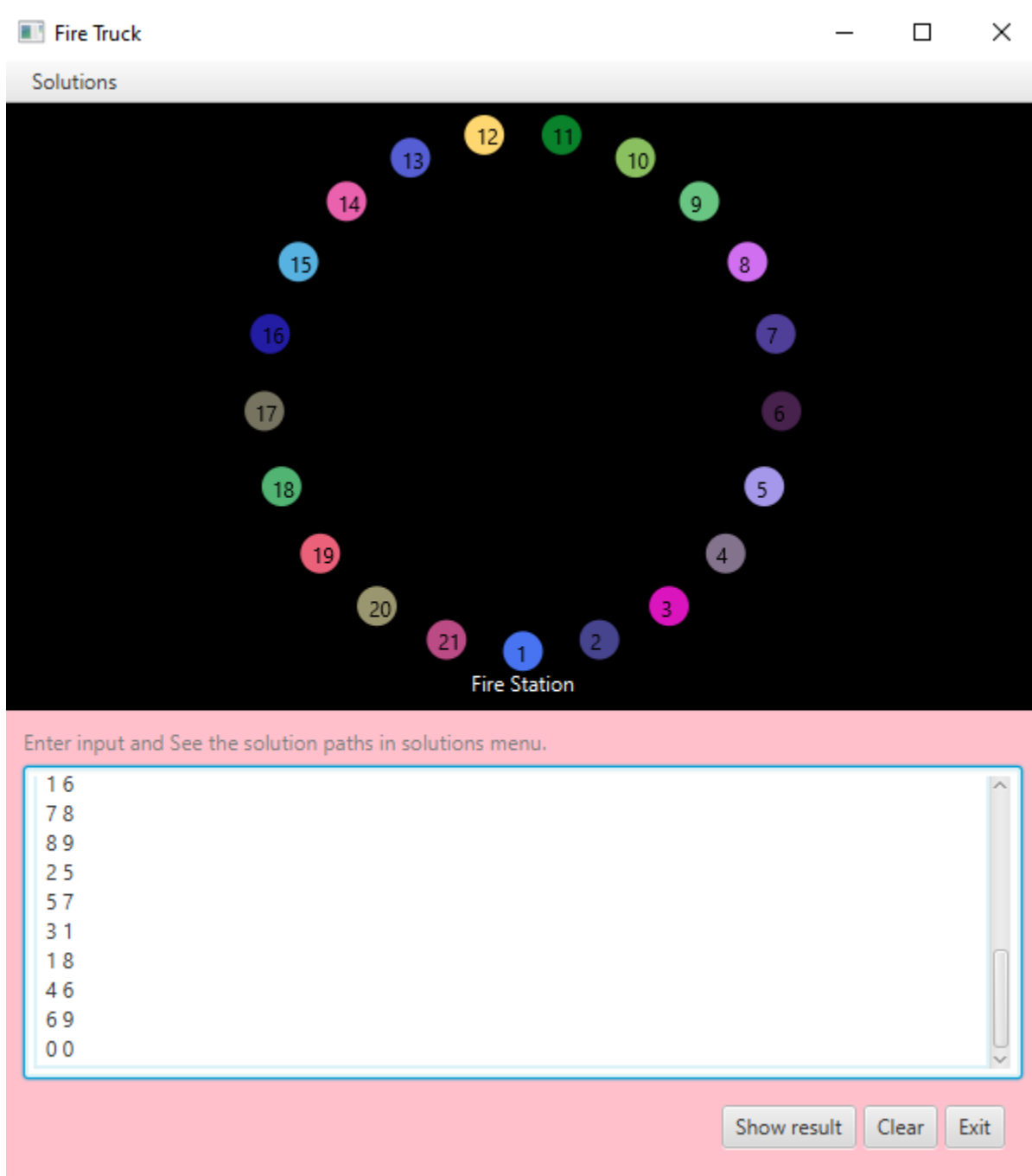


بنام خدا

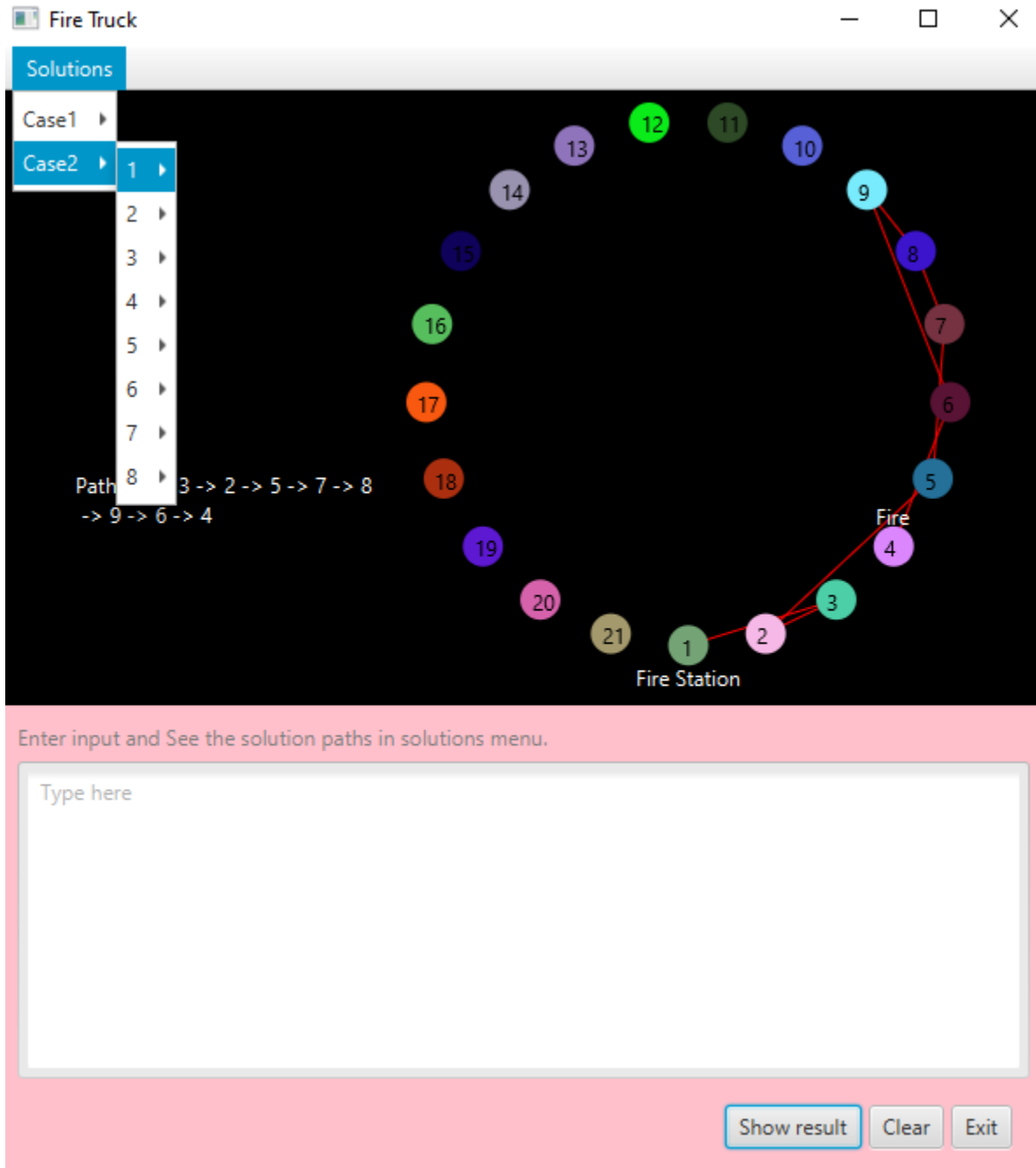
گزارش پروژه ی درس طراحی الگوریتم

نمونه ای از اجرای برنامه :

گرفتن ورودی :



دیدن نتایج :



روند اجرای برنامه و تحلیل پیچیدگی زمانی :

در این سؤال برای محاسبه ی تعداد مسیر های ممکن از آتش نشانی تا محل آتش سوزی با توجه به ورودی های برنامه و تقاطع هایی که مسدود نیستند از روش عقبگرد استفاده می کنیم و استفاده از تابعی به عنوان calcPath تمامی مسیر های ممکن را استخراج می کنیم و نمایش می دهیم.

آرایه ی ist برای ذخیره سازی تقاطع های باز استفاده شده است که در صورتی که تقاطع i باز باشد، مقدارخانه ی [i][j] آرایه برابر یک می شود، آرایه ی mv مسیر حرکت یا اینکه از چه تقاطع هایی می توانیم استفاده کنیم را نشان می دهد، pos مکان فعلی ما در مسیر و fire مکان آتش سوزی را نشان می دهد، mvNo تعداد تقاطع هایی که استفاده شدند را نشان می دهد.

Arraylist solution برای نگهداری مجموعه ی جواب ها استفاده شده است، در صورتی که pos = fire یعنی به مکان آتش سوزی رسیده باشیم مسیر را طی شده را که در آرایه ی mv ذخیره شده است به مجموعه ی جواب ها اضافه می کنیم، در صورتی که pos != fire باشد، دونه دونه با استفاده از ist چک می کنیم که کدام تقاطع باز است، بعد از آن که متوجه شدیم تقاطع باز است باید چک شود که قبلا انتخاب نشده باشد تا در یک حلقه گیر نیفتیم، بعد از آنکه مطمئن شدیم در یک حلقه نیفتادیم، تقاطع باز را به آرایه ی mv اضافه می کنیم و مکان فعلی خود را بروز رسانی می کنیم و دوباره تابع را فراخوانی می کنیم تا به محل آتش سوزی برسیم.

در حالت کلی برحسب اینکه از چه تعداد گره برای رسیدن به مقصد استفاده می کنیم در بدترین حالت پیچیدگی زمانی بصورت زیر است که بدلیل هرس شدن درخت فضای حالت در الگوریتم عقبگرد بهتر ازین مقدار است. (n تعداد گره هاست).

$$C(0,n) + C(1,n) + C(2,n) + \dots + C(n,n) = 2^n$$