گزارش پروژه **MAZE**

طاهره فهیمی 9539045

استاد درس : دکتر سارا بوربور

تدریس یار درس : محمد عصاری

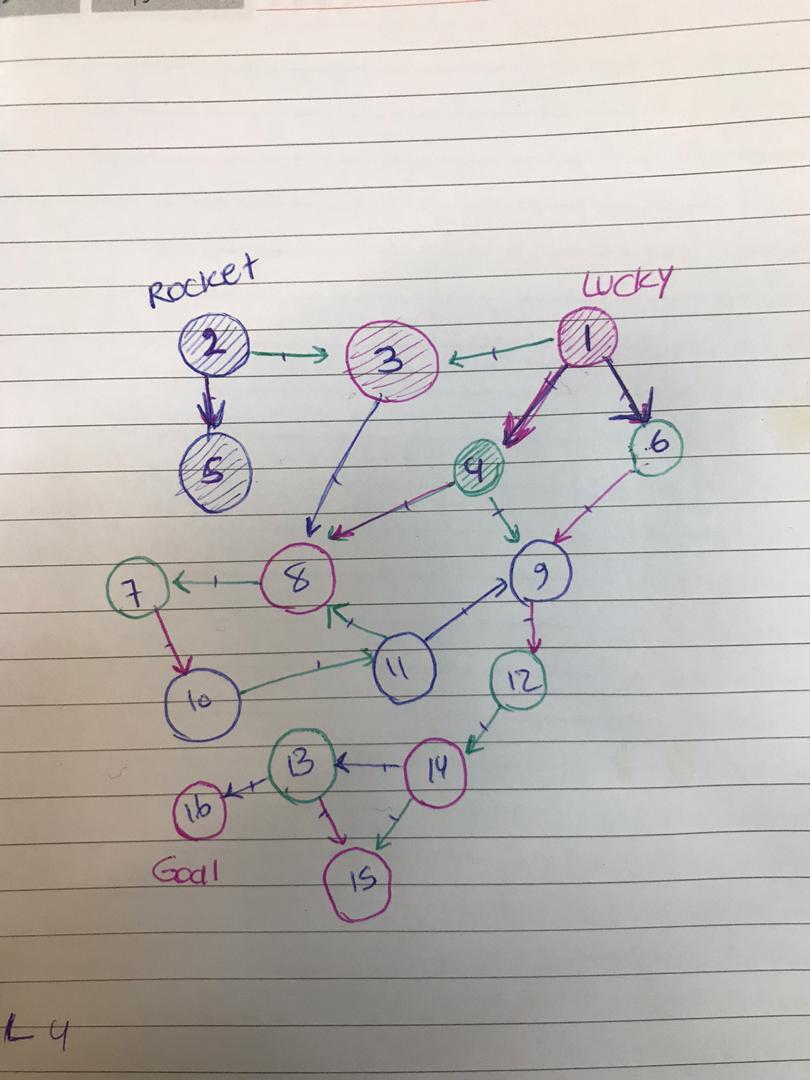
Problem Modeling:

بخش اول)

برای حل این مساله یک گراف جهت دار باید بسازیم که نودهای این گراف همان room ها و یال های جهت دار این گراف همان سالن ها(corridor) هستند علت استفاده از گراف جهت دار در این است که در تعریف مساله وقتی با یک corridor وارد یک room می شویم دیگر نمی توانیم از همان corridor به خانه ای که از ان امده بودیم برگردیم. علاوه بر این باید برای هر نود و هر یال رنگی را در نظر گرفت زیرا در صورتی که یک یال فعال نباشد امکان عبور از ان ممکن نیست.

بخش دوم)

برای بررسی بهتراین بخش، گراف زیر را در نظر میگیریم



بخش سوم)

باید یک مسیر از حداقل یکی از خانه های شروع تا هدف پیدا کنیم، برای این کار می توانیم از BFS یا DFS استفاده کنیم اما چون یال ها و نودها رنگ های خاصی دارند باید یک سری شرایط خاص هم به مساله اضافه کنیم.اما مساله در حالت ریلکس شده با BFSی خاصی دارند باید یک سری شرایط خاص هم به مساله اضافه کنیم.اما مساله در حالت ریلکس شده با BFS قابل حل است.

بخش چهارم)

فرض کنیم که مساله ما نتوانسته است برای یک مساله ای که پاسخ دارد جواب مورد نظر را پیدا کند پس باید در همه حالات رنگ دو خانه lucky و rocket با رنگ یال هایشان متفاوت باشد، اما اگر این نتیجه برقرار باشد یعنی مساله جواب نداشته است. پس تنها در حالتی که مساله جواب نداشته باشد الگوریتم ما نمی تواند مسیری را به عنوان مسیر رسیدن تا هدف بازگرداند.

مثلا گراف بخش دوم را اگر بررسی کنیم: چون rocket در خانه 2 که ابی رنگ هست قرار دارد پس lucky می تواند به خانه 6 با یال ابی رنگ برود. حال rocket به خانه 3 با یال سبز رنگ می رود زیرا lucky در خانه سبز رنگ قرار دارد بدین منوال lucky و rocket حرکت میکنند و تا زمانی که بن بست یا هدف نرسیم ادامه می دهیم.

Code Submission:

کد به فایل گزارش پیوست شده است.

در ضمن تمامی مراحل و متغیر ها به صورت کامنت توضیح داده شده اند.

Result:

برای ورودی داده شده در صورت پروژه خروجی به صورت زیر خواهد بود

L 7

R 10

L 11

R 15

L 6

R 14

L 2

R 8

L 7

R 4

L 11

R 1

L 6

R 10

L 2

R 15

L 7

R 14

L 11

R 8

L 6

R 4

L 2

L 3

R 1

L 12

R 10

L 17

R 15

L 16

R 14

R 20

L 14

R 21

L 8

R 22

L 4

R 23

L 1

R 27

L 10

R 26

L 15

R 25

L 14

L 20

R 28

Extra Credit:

این پروژه را می توانستیم با DFS ویا BFS حل کنیم.همان طور که می دانیم تنها عیبی که BFS در مقابل با DFS دارد این است که تعداد نودهای زیادی را نگهداری می کند اما برای حل این مساله اگر از BFS استفاده کنیم راه حل بهینه ای را بدست اورده ایم زیرا تعداد نودها خیلی زیاد نیستند پس حافظه زیادی را اشغال نمیکنند علاوه بر ان زمان رسیدن به هدف در این حالت کوتاه تر از DFS است.