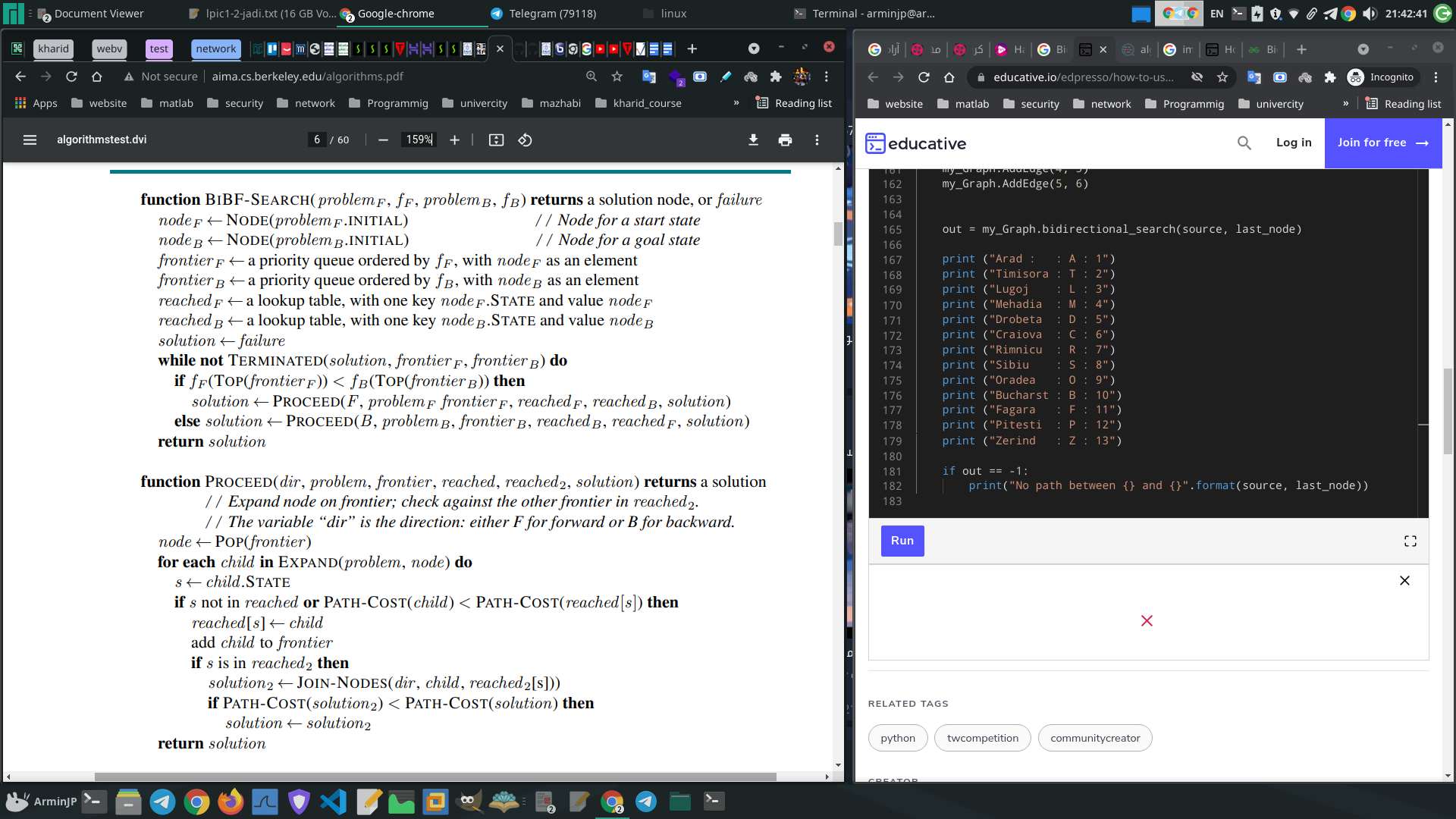
ــ به نام خدا ــ

| UI-AI-2021-3620020-03-**Quiz1** | | **کد تکلیف** | |
| --- | --- | --- | --- |
| **9** | | **شماره گروه** | |
| **CompuBrain** | | **نام گروه** | |
| **1400/07/18** | | **تاریخ تحویل** | |

| **اقدامات در این تمرین** | **شماره دانشجویی** | **نام و نام خانوادگی** | |
| --- | --- | --- | --- |
| **جمع آوری مطالب ، پیاده سازی و شخصی سازی کد ، تهیه ی داکیومنت** | **973613013** | **آرمین جعفرپیشه** | **1** |
| **جمع آوری مطالب** | **973613087** | **محمد نصر اصفهانی** | **2** |
| **جمع آوری مطالب** | **973613047** | **نیما عبدپور** | **3** |

عموما روشهای حل سوالات با رویکرد searching در یک جهت انجام می‌شوند که از سمت راس اولیه یا initial تا راس هدف یا goal می‌باشد ، اما در bidirectional search از دو راس برای جستجوی همزمان استفاده می‌شود . هدف این الگوریتم پیدا کردن کوتاه ترین مسیر از راس اولیه تا راس هدف می‌باشد.

شبه کد مربوط به این روش را در ادامه قرار داده ایم :



همانطور که در طول درس نیز فرا گرفتیم به طور عمومی حل سوالات با رویکرد search با فرستادن دو آرگومان به تابع که شامل : نوع مسئله و استراتژی می‌باشد ، صورت می‌پذیرد ، که خروجی نهایی می‌تواند یا راه حل نهایی و یا خطایی باشد که نشان از این موضوع می‌تواند داشته باشد که استراتژی انتخابی نتوانسته است راه حل مناسبی برای مسئله ی مطرح شده ، پیدا کند.

روش bidirectional هم از این قاعده مستثنی نیست با این تفاوت که چون از دو سمت حرکت برای حل مسئله صورت می‌پذیرد ، لذا دو آرگومان مربوطه برای هر دو طرف باید به شکل جداگانه مشخص و برای تابع ارسال شوند.

همچنین باید گفت در روش bidirectional ، استراتژی هایی که می‌توانند استفاده شوند هر کدام از روشهای سرچ ناآگاهانه ای که در طول درس فرا گرفتیم مثل BFS , DFS , IDS می‌توانند باشند.

ما برای پیاده سازی و بررسی این روش از استراتژی ای که در BFS دنبال می‌شد ، استفاده کردیم .

در کد پایتونی که پیاده سازی آن را انجام دادیم ، بخشی از کد را از این [لینک](https://www.geeksforgeeks.org/bidirectional-search/) کمک گرفتیم و در ادامه کد را شخصی سازی کرده و مسئله ی رومانی را در آن وارد کردیم که بتوانیم رویکرد پیاده سازی ای که انجام دادیم را در قبال مسئله ی رومانی مشاهده کنیم و ببینیم این مسئله به کمک روش bidirectional search و استفاده از استراتژی BFS آیا قابل حل هست یا خیر.

همچنین لازم به ذکر است که در کد پیاده سازی شده دو حالت سرچ وابسته به دو طرفه بودن روش رخ اتفاق می‌افتد ، Forward search که در آن از راس اولیه به سمت راس هدف حرکت می‌کنیم و Backward search که در آن از راس هدف به سمت راس اولیه در حرکت هستیم ، و در نهایت محل تلاقی این دو با هم نشان از اتمام فرایند search خواهد داشت . که البته اگر استراتژی ما درست انتخاب نشده باشد ممکن است جوابی این تلاقی اتفاق نیفتد و راه حلی یافت نشود.

همچنین مقصد اولیه ی ما شهر Arad با شماره ی ۱ و مقصد نهایی ما شهر Bucharest با شماره ی ۱۰ می باشد که تعیین این موارد از ملزومات روش دو طرفه می‌باشد ، به علاوه شهرهایی در نقشه که از بخارست به بعد می‌باشند را در این مسئله در نظر نگرفتیم چرا که در فضای بررسی این سوال جایی نداشتند.

در جدول زیر به هر کدام از شهرهایی که در طول مسیر قرار دارند ، عددی نسبت دادیم ، تا بر مبنای این اعداد ، در پیاده سازی خود جلو برویم. در نهایت اگر راه حلی پیدا شود ، مسیری که برگردانده می‌شود ، بر مبنای این اعداد خواهد بود ، که به کمک جدول زیر می توان نام شهر ها را از آن واکشی کرد.

| شماره ی شهر | حرف اول | نام شهر |
| --- | --- | --- |
| **1** | **A** | **Arad** |
| **2** | **T** | **Timisora** |
| **3** | **L** | **Lugoj** |
| **4** | **M** | **Mehadia** |
| **5** | **D** | **Drobeta** |
| **6** | **C** | **Craiova** |
| **7** | **R** | **Rimnicu** |
| **8** | **S** | **Sibiu** |
| **9** | **O** | **Oradea** |
| **10** | **B** | **Bucharst** |
| **11** | **F** | **Fagara** |
| **12** | **P** | **Pitesti** |
| **13** | **Z** | **Zerind** |

کد پیاده سازی شده در پیوست فایل داکیومنت با نام Quiz1.py قرار داده شده است همچنین فایل اجرایی که در محیط ترمینال لینوکس قابل اجرا می‌باشد نیز در همین پیوست با نام Quiz1.ext قرار داده شده است.

خروجی کد پیاده سازی شده پس از اجرا نیز به صورت زیر می باشد :

| The Number Of city is :  Arad : : A : 1 Timisora : T : 2 Lugoj : L : 3 Mehadia : M : 4 Drobeta : D : 5 Craiova : C : 6 Rimnicu : R : 7 Sibiu : S : 8 Oradea : O : 9 Bucharst : B : 10 Fagara : F : 11 Pitesti : P : 12 Zerind : Z : 13  Path exists between 1 and 10 Intersection at : 7 \*\*\*\*\*Path\*\*\*\*\* 1 8 7 12 10 |
| --- |

همانطور که مشخص می‌باشد با استراتژی ای که انتخاب کردیم ، برای مسئله راه حلی برگردانه شده است و مسیری که باید پیموده شود نیز در آن ذکر شده است. پس bidirectional search با استراتژی انتخابی ما قادر است مسئله ی طی کردن مسیر از شهر آراد به بخارست را برای ما حل کند.