ماژول گراف :

در این فایل کلاس های مورد نیاز گراف پیاده سازی شده است . گراف توسط روشadjacency map پیاده سازی شده است . در واقع هر node توسط یک dictionary که کلید های ان node های همسایه و مقدار ان کلید ها edge متصل کننده است ، مقدار دهی می شود و همه ی node ها در یک لیست ذخیره می شوند.مسافت ، قیمت پرواز و زمان پرواز در کلاس edge ذخیره می شود و بقیه ی اطلاعات در کلاس node ذخیره می شود . همچنین توسط تابع checkValues بررسی می شود که اطلاعات node ها تکراری نباشد .

تابع getVertex ، یک رشته از نام فرودگاه میگیرد و node ان را بر میگرداند.

ماژول a\_star :

این فایل مربوط به پیاده سازی الگوریتم a star است . تابع gn ، g(n) مربوط به node ها را محاسبه می کند که مقدار ان برابر با مجموع مسافت و دو برابر قیمت پرواز بین دو فرودگاه است. تابع heuristic ، h(n) مربوط به node ها را محاسبه میکند که مقدار ان برابر با مسیر مستقیم بین دو فرودگاه است . این کار با استفاده از دو متد latitude و longitude انجام می شود . (برای محاسبه مسیر مستقیم از سایت geeksforgeeks استفاده شده است)

تابع a\_star\_search :

این تابع دو فرودگاه میگیرد و بهترین مسیر را براساس الگوریتم a\_star ذخیره میکند . به این صورت که ابتدا node مربوط به این دو فرودگاه را ذخیره میکند . سپس لیست frontier برای بررسی node ها ، لیست visited برای جلوگیری از بررسی مجدد ، دیکشنری distance برای نگهداری مقادیر f(n) و دیکشنری answer برای نگهداری بهترین مسیر پرواز تعریف می کند.

ابتدا داخل frontier فقط مبدا قرار دارد و در اجرای اول انتخاب می شود . سپس مقدار f(n) ان در distance قرار میگیرد . سپس هزینه تمام همسایه های ان (که در واقع برابر است با مجموع هزینه مصرف شده تا ان لحظه با g(n) ) حساب شده و داخل distance قرار میگیرد . همچینین همه ی node ها هم داخل frontier ذخیره می شوند . در اخر هم node مبدا از لیست frontier حذف و به لیست visited اضافه میشود .

در اجرای بعدی ، برای هر node مجموع هزینه ی مصرف شده تا ان لحظه و h(n) ان node محاسبه می شود و کمترین ان انتخاب می شود . سپس مثل قبل ، همسایه های node انتخاب شده به داخل frontier وارد میشوند و هزینه ی ان ها هم داخل distance قرار میگیرد . در اخر هم node انتخاب شده از frontier حذف میشود و به visited اضافه می شود .

در بررسی همسایه های node انتخاب شده ، اگر هزینه دسترسی به یک node که قبلا از طریق یک مسیر دیگر بررسی شده است ، با استفاده از این مسیر کمتر شود ، مقدار هزینه ان اپدیت میشود، همچنین دوباره به frontier اضافه میشود که دوباره مورد بررسی قرار گیرد .

در هر اجرا ، همسایه های کمترین node انتخاب شده به دیکشنری answer اضافه می شود که value ان برابر است با کمترین node انتخاب شده . با این کار ، مسیر رسیدن به مقصد مشخص می شود .

ماژول time :

در این ماژول execution time برای الگوریتم ها محاسبه می شود . به این صورت که تایم شروع در متغیر start\_time و تایم پایان در end\_time ذخیره میشود . تفاوت این دو متغیر برابر با زمان طول کشیده بر حسب ثانیه است .