به نام خدا

کیمیا قدسی فر -HW2-9623088-

سوال 1:

در این سوال اینکه کدام بخش سوال را قرار است اجرا کنیم ازکاربر پرسیده می شود. ابتدا ازکاربر می پرسیم که بخش الف اجرا شود یا بخش ب.

کاربر برای اجرای الف ، 1 و برای اجرای ب ، 2را وارد میکند.

بعد در تابع find Route با توجه به مقدار داده شده از روش بخش اول یا دوم برای تعیین مسیر استفاده می کنیم.

برای اجرای بخش ج ازکاربر پرسیدیم که تمایل به پیدا کردن مسیر بهینه دارد یا نه.

برای نوشتن کد این سوال یک classبه اسم map تعریف شده که توابع زیر را دارد و در هر بخش کارکرد آن توابع توضیح داده می شود:

ـ تابع Constructor : ارتفاع هر ناحیه رندوم مشخص می شود.

در نقشه به تک تک نقاط "ـ" نسبت داده می شود.

ـ تابع Destructor: حافظه هایی که داینامیک بودند و برای نمایش نقشه و ارتفاعها اشغال کرده بودیم پاک می کند.

–تابع showMap: برای نمایش نقشه با مقادیر است.

ـ تابع showRoute: برای نمایش نقشه با علایم است.

ـ تابعfindRoute:با توجه به guide که ازکاربرگرفتیم بخش الف را اجرا می کنیم یا بخش ب. (شرط اول حالت الف و شرط دوم حالت ب)

در هر یک از حالات الف یا ب:

یک مختصات برای حرکتمان مثل y وx در نظر میگیریم .که ابتدا 0 است.

سپس بررسی می کنیم که به دیوار و انتهای محیط حرکت نخورده باشیم.

بعد قدرمطلق نقاطی که امکان حرکت به آنها را داریم بررسی کرده ، مسیر را انتخاب می کنیم. برای حرکت به راست x را باید افزایش داد و برای حرکت به پایین Yرا. برای حرکت به راست و پایین هر دو آنها.

پس از مشخص شدن مختصات نقطه بعد ، آن نقطه را به کمک + نشان گذاری می کنیم.بعد شرط رسیدن به دیوار عمودی یا افقی را بررسی می کنیم که در این صورت مستقیم باید ادامه داد.

در هر مرحله اختلاف ارتفاع ذخیره شده و بعدها به عنوان Distance

نمایش داده می شود.

ـ تابع findOptimized: برای یافتن بهترین مسیر ممکن است.

اگر پایین آمدن را با 1 و راست رفتن را با 0 نشان بدهیم ونقشه n*n باشد آنگاه باید n-1 بار راست و n-1بار پایین بیاییم. بنابراین اگر یک عدد 2(n-1) رقمی داشته باشیم که متشکل از تعدادمساوی 0 و 1 باشد و پایین آمدن را با 1 و راست رفتن را با 0 نشان بدهیم آنگاه هر جایگشت این عدد یک مسیر را نشان خواهد داد. به همین منظور یک vector به نام permute تعریف کردیم که قرار است نقش همان عدد را بازی کند.به همین دلیل n-1 نود به 0 و n-1 نود به 1 اختصاص میدهیم.

تعداد این جایگشت ها عبارت است از :

2(n-1)!/(n-1)!(n-1)!

بنابراین در یک حلقه به همین تعداد عدد متشکل از 0 و1 را جایگشت می دهیم و در هر حالت مسافت طی شده را ذخیره کرده و به یک آرایه از مسافتها انتقال می دهیم تا بدانیم با چند جایگشت به کمترین مسافتها

ممکن رسیدیم.برای رفتن مسیر طبق هر یک از permute و محاسبه مسافت از تابع comp استفاده خواهیم کرد.این تابع مشابه بخش الف و ب اما طبق جایگشت اعلام شده کار می کند.

با مقایسه مقادیر distance به دست آمده و پیدا کردن کمترین انها میتوان فهمید پس از چند جایگشت به کمترین مسافت رسیدیم.

تابع go تابعی است که به کمک آن مسیر پیدا شده را می رویم.

سوال 2 :

از هر نوع کلاس یک object می سازیم و یک اشاره گر به هریک از آنها.حال برای اعداد 1و10و....10^6 تابع runTime را با کمک اشاره گر به تابع و اشاره گر به object صدا میزنیم.

این تابع زمان قبل و بعد اجرای هر یک از توابع را اندازه گرفته و از هم کم می کند.

سوال3:

یک کلاس به اسم old تعریف می کنیم که اطلاعات هر یک از تراکنش شامل روز ، کد محصول ، کد مشتری در آن ذخیره می شود.

برای خواندن اطلاعات فایل و پردازش آنهانیاز به ذخیره اطلاعاتش داریم.

یک vector از old ها می سازیم.هر خط فایل را خوانده ، بخش بخش می کنیم و روز ، کد محصول وکد مشتری را در هر نود ذخیره می کنیم.

سپس در vector حرکت می کنیم.روزهای مشابه را شناسایی می کنیم.

برای حرکت در طول vector از start , end استفاده می کنیم.

با تنظیم آنها در طول یک روز حرکت کرده و متمایز بودن هر یک از محصولات یا مشتری ها را چک کرده ، جمع می کنیم و سپس چاپ می کنیم.