



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

دانشکده مهندسی برق

درس برنامه نویسی پیشرفته

استاد درس: دکتر جهانشاهی

تمرین سری دوم

نام و نام خانوادگی: علیرضا خیاطیان

شماره دانشجویی: 9523039

## سوال اول

### روند برنامه (الگوریتم)

این سوال شامل 3 بخش است. برای حل آن یک کلاس به نام Map تعریف می کنیم که شامل توابع نمایش نقشه و نمایش مسیر و یافتن مسیر و یافتن کوتاه ترین مسیر بخش اول است که به تفصیل توضیح داده می شود

ابتدا یک آرایه دینامیک دو بعدی به نام arr می سازیم و آن را با اعداد رندم 0 تا 100 پر میکنیم و سپس آنها را چاپ می کنیم در نهایت آن را در distractor حذف می کنیم

### بخش اول

با مقایسه فاصله ها در جهت راست و پایین در هر مرحله کوتاه تر را انتخاب کرده و هر خانه که جزو مسیر شد را با گذاشتن عدد 1 در arr\_show تعیین میکنیم و بقیه را صفر قرار میدهیم توجه داریم برای دسترسی سراسری به این آرایه جهت چاپ در تابع showRoute آن را به عنوان یک جز کلاس به صورت آرایه دینامیک تعریف می کنیم که در نهایت distractor حذف می کنیم به علاوه با گذاشتن شرط و حلقه تعیین می کنیم در صورتی که به دیوار رسید تا نقطه پایانی مستقیم برود

### بخش دوم

در این بخش می خواهیم علاوه بر شرایط بخش اول حرکت به صورت مورب را نیز اضافه کنیم بدین جهت d3 را تعریف می کنیم که فاصله از عنصر قطری است حال باید d3 را با d2 که حرکت به سمت راست و d1 که حرکت به سمت پایین است مقایسه کنیم و کوچک ترین آن ها را بیابیم برای آن که تنها یک تابع findRoute برای هر دو بخش اول و دوم بنویسیم متغیر status را به عنوان ورودی به تابع findRoute می دهیم که موقع صدا زدن تابع در main در صورتی که ورودی A باشد منظور بخش اول و اگر ورودی B باشد منظور بخش دوم است میدانیم شرایط بخش دوم بر بخش اول محیط است بنابراین برای تفکیک شرایط دو بخش از هم یک شرط قرار می دهیم که اگر ورودی A باشد d3 برابر مجموع d1 و d2 قرار میدهیم که شرط های اضافی بخش دوم در بخش اول مشکلی ایجاد نکند

### بخش سوم

در این بخش می خواهیم کوتاه ترین مسیر ممکن طبق شرایط بخش اول یعنی حرکت به سمت راست و پایین را بیابیم برای این مهم از الگوریتم زیر استفاده می کنیم

میدانیم در یک نقشه  $n \times n$  باید  $n/2$  خانه به سمت راست و  $n/2$  خانه به سمت پایین برویم اما ترتیب حرکت را نمیدانیم به طور کلی برای انتخاب با حفظ اهمیت ترتیب انتخاب ها  $\text{pow}(2, n)$  حالت وجود دارد اگر بیابیم به فرض حرکت به سمت راست برابر 1 و حرکت به سمت پایین را 0 در نظر بگیریم از این حالات آن حالاتی مورد قبول است که مجموع 1 های آن برابر  $n/2$  باشد بنابراین با قرار دادن یک شمارنده به نام i و تبدیل آن به باینری و بررسی صحت شرط فوق تمامی حالت های ممکن حاصل می شود سپس با مجموع فاصله ها را برای هر مسیر را مقایسه کرده . کوتاه ترین مسیر را می یابیم و در نهایت به کمک showRoute چاپ می کنیم

## سوال دوم

### روند برنامه (الگوریتم)

دو کلاس به نام های libArr و libVec ایجاد می کنیم و داخل آن هر یک تابع counter را ایجاد می کنیم که به ترتیب یک آرایه و وکتور داخل تابع ایجاد می کنیم و آن را داخل یک حلقه به طول ورودی می دهیم که مقادیر هر خانه را برابر مقدار شماره حلقه کند سپس به main رفته و یک object از هر کلاس میسازیم به علاوه یک اشاره گر به تابع کلاس برای هر کلاس می نویسیم حال اشاره گر به تابع و object را به تابع runtime میفرستیم

### RunTime

می خواهیم با استفاده از توابع موجود در chrono library مدت زمان اجرا را بیابیم برای این مهم فرایند انجام را مابین start, stop قرار داده و مدت زمان را از اختلاف این دو با استفاده از duration محاسبه می کنیم نکته قابل توجه اینکه برای محاسبه از nanoseconds استفاده می کنیم که در نهایت با تقسیم به 1000000 تبدیل به milliseconds میکنیم تا بتوانیم با دقت بیشتر و مقادیر کمتر از 1 میلی ثانیه را محاسبه کنیم

حال پس از دریافت ورودی که یک اشاره گر به تابع کلاس و object آن است تابع counter را صدا می کنیم و مقدار مجموع را چاپ می کنیم

## سوال سوم

### روند برنامه (الگوریتم)

ابتدا یک فایل db.txt را می خوانیم و محتوای آن را در 4 string به نام های day time product\_id customer\_id ذخیره می کنیم یک vector از نوع آرایه ای از string ها ایجاد می کنیم و به این در ترتیب کل data base را در یک vector ذخیره می کنیم حال یک vector به نام separator ایجاد می کنیم میدانیم در این data base روز ها به ترتیب اند بنابراین برای جداسازی اطلاعات روز ها از هم separator را تعریف می کنیم که و شماره خانه هایی که روز آن با روز بعدیش متفاوت است را در آن ذخیره می کنیم پس از جداسازی اطلاعات یک روز از روز های دیگر اطلاعات مربوط به product\_id و customer\_id را در دو آرایه مجزا ذخیره می کنیم برای آن که تعداد تنوع کالا و مشتریان را بیابیم باید ابتدا آن ها را sort کنیم که در اینجا دو آرایه را به تابع selectionSort ارسال می کنیم حال پس از مرتب کردن این دئ آرایه آن ئو را برای شمارش تنوع کالا و مشتریان به تابع counterFunc میفرستیم و در دو آرایه دیگر ذخیره می کنیم در نهایت در فایل dbnew.txt ذخیره می کنیم.

## سوال چهارم

کتابخانه materialize اضافه می کنیم در بخش body یک بخش main ایجاد می کنیم که خود شامل سه بخش است که به ترتیب menu و second و badane می باشند در بخش badane دو بخش contact و chat را ایجاد می کنیم که به صورت افقی قرار می گیرند. در بخش contact بخش هایی با عنوان daroni ایجاد می کنیم که شامل search bar و contact\_content و سایر بخش ها می باشد. بخش chat نیز شامل دو بخش chat\_content و message می باشد به این ترتیب ساختار کلی شکل گرفته است

تمامی **contact** ها دارای **hover** خواسته شده در سوال می باشند به علاوه علامت کنار تلگرام دارای **modal** است که در صورت کلیک روی آن صفحه **setting** باز می شود برای این منظور نیاز به **jquery** بود که اضافه شد

برای آن که صفحه وب قابل استفاده در موبایل باشد یا به عبارتی دیگر **responsive** باشد غالب ابعاد را به صورت درصد بیان شد و برای **responsive** به طور کلی از دستور **materialize** با نام کلاس **row** و **col s6 m12** استفاده شد.

عکس ها نیز با آپلود در سایت **uupload.com** لینک های آن ها قرار داده شده اند .