دانشکده مهندسی برق

درس برنامه نویسی پیشرفته

استاد درس:دکتر جهانشاهی

تمرین سری دوم

نام ونام خانوادگی:علیرضا خیاطیان

شماره دانشجویی:۹۵۲۳۰۳۹

Front-End

در ابتدا بخشی با نام introduction ایجاد می کنیم تا ابعاد صفحه و نام بازیکن ها را بگیریم و یک کلید با عنوان Go قرار میدهیم که دارای getsize onclick می باشد با زدن کلید وارد صفحه بازی می شویم

تابع getId را درون get size تعریف می کنیم تعریف می کنیم که id یک event را برمی گرداند به علاوه تابع makeline را نیز تعریف می کنیم که بعد از چگونگی تولید بخش های صفحه بازی توضیح داده خواهد شد

ساختار صفحه به این صورت است یک بخش به نام gameکه شامل بقیه بخش هاست که عرض آن را با استفاده از عرض ورودی *۶,۷ و واحدvw تعیین می کنیم تا responsive باشد

در ابتدا به کمک یک حلقه روی ارتفاع یک ردیف از خط و نقطه های عمودی تشکیل می دهیم سپس درو ن این حلقه حلقه دیگری به اندازه عرض ورودی ایجاد می کنیم بخش اول که شامل خط و نقطه های افقی است برای این کار هر نقطه و خط را به ترتیب node و line می نامیم که هر دو این ها جزیی از بخش دیگری به نام one1 و هر یک از one1 ها جزیی از truckload می باشند که آن را اضافه می کنیم

سپس box ها به همین ترتیب اضافه می کنیم و سپس همانند خط های افقی خط های عمودی را اضافه می کنیم سپس از این حلقه تو در تو خارج شده و حلقه دیگر به طول عرض وارده شده ایجاد می کنیم که نقطه ای آخر هر ردیف افقی قرار دهیم و به علاوه ردیف افقی دیگر برای بسته صفحه میزنیم و در پایان این ردیف نیز نقطه ایی اضافه می کنیم

نکته قابل ذکر آن که تمام ابعاد را بر اساس vh و vw داده شده تا صفحف responsive باشد

حال به سراغ تابع makeline می رویم که خط را رنگی می کند و در صورتی که تمام خطوط پیرامون یک box رنگی باشد آن را بر اساس آخرین خطی که رنگی شده رنگی می کند در این تابع می خواهیم به کمک id خطوط بررسی کنیم که آیا خزوز پیرامون یک box رنگی هست یا خیر به طور کلی ما سه نوع id داریم خزوز افقی به صورت ((i-1)*length+j) و ردیف پایانی از خطوط افقی به صورت (length*height+j) به این طریق تمامی خطوط افقی با اعداد پشت سر هم مشخص میشوند

خطوط عمودی که جهت تمییز با خزوز افقی ابتدا آن ها حرف v اضافه شده و به صورت "v"+((i-1) *length+j) می باشد وخط عمودی ابتدایی به صورت "f"+((i-1)* length+1) می باشد

در ابتدا تابع makeline متغیر counter را به ابن صورت تعریف می کنیم که اگر زوج باشد خط آبی شود و اگر فرد باشد خط قرمز شود و در صورتی که فردی یک خانه را ببرد یک جایزه دارد و ان اینکه یک نوبت دیگر دارد که counter که در ابتدا تابع زیاد شده بود یکی کم می شود بنابراین دو شرط کلی داریم یکی برای آبی و یکی برای قرمز و درون این دو شرط، شرط های دیگری داریم که تعین می کند box رنگی بشود یا خیر

ابتدا مغیر r را که id ورودی است می بابیم سپس مغیر res0 و res1 و res2 را تعریف میکنیم با توجه به res0 ۳ شرط ایجاد می کنیم اگر برابر v باشد یعنی خط ورودی عمودی است اگر f باشد یعنی خط وروردی خط عمودی ابتدایی است و گرنه خط افقی است حال هر یک از این سه شرط را توضیح می دهیم

شرط خطوط افقی شامل ۴ بخش می شود بخش اول درصورتی که box مذکور بالای خط باشد بخش دوم اگر boxپایین خط باشد بخش سوم که برای خطوط افقی ریف اخر است و بخش چهارم که برای خط اول ردیف آخر است به عنوان نمونه بخش اول را توضیح میدهیم

دو متغیر با نام های test1و test2به ترتیب تعییه کننده idخط بالایی و خط پایینی برای ورودی و خطوط چپ و راست برای خطوط عمودی است در بخش اول برسسی می کنیم خط بالای خط افقی وروردی ایا رنگی است سپس برسی می کنیم خطوط چپ و راست بالای این خط آیا رنگی است در صورتی که هر سه خط داده شده رنگی باشند box را به رنگ خط چهارم در می آوریم

به همین ترتیب متغیر های test1_resو test2_res را برای خطوطی تعریغ می کنیم که دارای حرف vو f هستند

بخش عمودی نیز شامل ۳ بخش است اگرbox در راست خط وروردی باشد اگر box در سمت چپ ورودی باشد و بخش سوم اگر اگر ورودی جزو خطوط عمودی ابتدایی باشد که همانند بخش خطوط افقی و با کمک متغیر های توضیح داده شده بررسی می کند که آیا box رنگی شود یا خیر

بخش خطوط ابتدایی نیز همانند بخش خطوط عمودی است

حال ۲ متغیر به نام blue_playerوred_player ایجاد می کنیم که در صورتی که مربع رنگی شود به آن ها اضافه می شود

و در نهایت زمانی که تمام box ها رنگی شوند یعنی $red\_playey + blue\_player = height * length$ پیام پیروزی یا مساوی بر اساس تعداد مربع های رنگ شده برای هر طرف از بازی اعلام می شود.

سوال اول

روند برنامه(الگوریتم)

میخواهیم ساختمان داده MaxHeap را ایجاد کنیم بخش های مختلف کلاس Maxheap در زیر توضیح داده شده است

Constructor

کلاس شامل ۳ constructor است که عبارت اند از Default constructor, Copy constructor و حالت سوم که یک آرایه را به عنوان ورودی میگیرد

کلاس MaxHeap شامل ۳ متغیر است heap_size, capacity و harr. متغیر heap_size تعداد خانه های ساختمان داده را مشخص می کند .capacity ظرفیت ساختمان داده را تعین میکند و harr نیز آرایه مرتب شده ساختمان داده است

به دلیل وجود آرایه دینامیک harr وجود copy constructor الزامیست به این ترتیب که در copy constructor آرایه دینامیک جدیدی new می شود و مقادیر داخل آن با harr برابر قرار داده میشود دقت شود که نباید آرایه دینامیک جدید با harr برابر قرار داده شود زیرا double free رخ می دهد

توابع

add: در ابتدا capacity و heap_size یک واحد زیاد شده سپس مقدار ورودی به انتهای harr اضافه می شود و در نهایت harr مرتب میشود

left & right: این دو تابع شماره خانه های فرزند راست و چپ را برمیگرداند

leftChild & rightChild: این دو تابع مقدار خانه های فرزند راست و چپ را برمیگرداند

Maxheapify : این یک تابع بازگشتی است که درخت را مرتب میکند به این ترتیب که شماره ریشه ای که میخواهیم از آن به بعد مرتب کنیم را میگیرد و با استفاده از swap آن را مرتب می کند

Parent :که شماره والد را برمی گرداند

parent :که مقدار دالد را برمی گرداند

Delete :این تابع مقدار ماکزیمم یا همان ریشه را از درخت حذف می کند به این ترتیب که ابتدا مقدار ریشه خذف میشود سپس مقدار اخر تابع در مقدار ریشه ذخیره میشود و heap_size یک واحد کم میشود حال باید کل درخت مرتب شود به همین دلیل تابع Maxheapify با وروردی صفر صدا زده می شود

printArray : درخت را با رعایت طبقه بندی چاپ میکند میدانیم که تعداد خانه ها تا هر طبقه 2^n+1 می باشد

getHeight : تعداد طبقات را با توجه به تعداد خانه ها تا هر طبقه که 2^n-1 است را چاپ میکند

Max : که شماره ریشه یا همان بیشترین مقدار را برمیگرداند

Oprator [i] :مقدار شماره ورودی را برمیگرداند

Oprator<< : که Maxheapرا به صورت درخت چاپ می کند.چون شی ای که این عملگر روی آن اعمال میشود از جنس ofstream است در خارج از کلاس تعریف می شود به علاوه چون ورودی تابع از جنس Maxheap است که هنوز تعریف نشده از Forward declaration استفاده شده است

extractMax :این تابع همانند تابع delete عمل می کند با این تفاوت که بیشترین مقدار را برمیگرداند این تابع در Heapsort کاربرد دارد

Heapsort : این تابع در هر مرحله بشترین مقدار درخت باقیمانده را به کمک تابع extractMax به دست می آورد و Maxheap مرتب شده با این الگو را در شی ای که این تابع روی آن اعمال شده ذخیره میکند به این ترتیب دیگر دسترسی به ساختمان قبلی این شی وجود ندارد

Oprator+ :

Oprator- :

Oprator= :

سوال دوم

روند برنامه(الگوریتم)

در این سوال کلاسی مشابه vector به نام Vec ایجاد می کنیم . این کلاس دارای Default Constructor، Copy constructor و Move constructor می باشد که در زیر توضیح داده میشوند

عناصر کلاس عبارت اند از capacity ,size,arr که publicهستند تا بتوان طبق خواسته سوال درون main به آن ها دسترسی داشت به علاوه دو عنصر Len,capacity_middle که در توابع کاربرد دارند نیز به صورت private تعریف شده اند

Copy Constructor لازم است تا از double free جلوگیری کنیم .یک آرایه دینامیک جدید ایجاد می کنیم و مقادیر arr را در آن ذخیره می کنیم اما در Move Constructor در صورتی که ورودی rvalue باشد صدا زده می شود به صورت که اصطلاحا arr مربوط به ورودی است که از جنس rvalue است را میدزدیم و در arr شی ذخیره می کنیم و در نهایت نیز arr ورودی rvalue را nullptr قرار می دهیم .تفاوت اصلی Copy و Move آن است که در Move آرایه دینامیک جدید ایجاد نمیشود و از همان ارایه ورودی ریاrvalue که به هر جهت از بین می رود استفاده میشود اما در copy ارایه دینامیک جدیدی ایجاد می شود

Push_back : در این تابع می خواهیم یک عنصر دیگر به انتهای وکتور اضافه کنیم برای این کار ابتدا size را یک واحد افزایش میدهیم حال جز صحیح لگاریتم ۲ size را محاسبه و در capacity_middleذخیره می کنیم به این ترتیب در صورتی که capacity قبلی جوابگو نباشد آرایه دینامیک جدیدی ایجاد می کنیم و مقادیر آرایه دینامیک arr قبلی را به اضافه مقدار جدیدی که میخواهیم اضافه کنیم در آن ذخیره می کنیم به این ترتیب به تعریف capacity_middle کلاس بهینه تر شده و دیگر در مواردی که capacity قبلی جواب گو هست آرایه دینامیک جیدی ایجاد نمیشود.

$$\begin{aligned}\vec{A}\times\vec{B} &= (A_yB_z - A_zB_y)\,\hat{\imath} \;+\\ &\quad (A_zB_x - A_xB_z)\,\hat{\jmath} \;+\\ &\quad (A_xB_y - A_yB_x)\,\hat{k}\end{aligned}$$

تذکر:دقت شود که در صورت سوال خواسته شده در هر بار push و pop آرایه دینامیکی ایجاد و آرایه قبلی حذف شود که تابع pop به این صورت عمل می کند اما اگر بخواهیم کلاس بهینه تر عمل کند متغیر capacity_middle را تعریف کرده که تنها درصورت جوابگو نبودن capacity قبلی آرایه جدیدی ایجاد شود

Pop_back : در این تابع می خواهیم یک عنصر از انتهای وکتور حذف کنیم برای این کار ابتدا یک واحد از size کم می کنیم سپس به محاسبه capacity همانند روشی که در push_back توضیح داده شد می پردازیم و آرایه جدیدی به طول capacity ایجاد می کنیم و مقادیر arr قبلی را به جز آخرین مقدار آن را در آرایه جدید ذخیره می کنیم و در نهایت آرایه قبلی را nullptr می کنیم

Show : در این تابع مقادیر ذخیره شده در arr را به ترتیب چاپ می کنیم

Dot : در این تابع می خواهیم حاصل ضرب نقطه ایی دو بردار را محاسبه کنیم در صورتی که اندازه دو بردار برابر نباشد تابع error می دهد

برای محاسبه هر دو عنصر متناظر از دو بردار را ضرب کرده و حاصل آن ها را با هم جمع می کنیم و در نهایت چاپ می کنیم

Cross : میخواهیم حاصل ضرب خارجی دو بردار را محاسبه کنیم برای این کار ابتدا چک می کنیم که اندازه هر دو بردار برابر ۳ باشد سپس با توجه به رابطه ریاضی محاسبه بردار حاصل از ضرب خارجی آن را محاسبه و چاپ می کنیم

Operator*:همانند cross عمل می کند با این تفاوت که خروجی نداشته و بردار حاصل را تنها چاپ میکندو به صورت operator عمل می کند

Operator< = >: operator==و operator< را بر اساس مقایسه اندازه بردار ها نوشته و با اضافه کردن کتابخانه utility سایر حالت ها از این دو حالت پوشش داده می شود

Operator=: میدانیم هر کلاسی که دارای آرایه دینامیک باشد به operator= نیاز دارد تا از double free جلوگیری شود برای این کار ابتدا arr شی را حذف کرده و سپس مقادیر arr وردی را در آن ذخیره می کنیم و در نهایت شی را برمیگردانیم نکته جالب برای جلوگیری از copy constructor خروجی به ورت reference می باشد.

سوال سوم

روند برنامه(الگوریتم)

این سوال شامل دو کلاس human و oracle می باشد که به ترتیب توضیح داده می شوند

Human

متغیر های این کلاس عبارت اند از: نام،نام خانوادگی،رنگ مو،رنگ چشم،سن،جنسیت و تعداد فرزندان به علاوه ۳ مغیر از جنس Human* که به پدر،مادر و همسر هر فرد بستگی دارد

مشخصات فردی هر شخص را private نعریف کرده و به همین دلیل توابع get را برای دستابی به آنها از main مینویسیم

Constructor

با توجه به این که کلاس دارای آرایه دینامیک هست copy constructor و = operator نیاز است

این کلاش شامل ۳ constructorاست حالت عادی که در صورت سوال آمده default constructor و copy constructor

در constructor اول مقادیر داده شده را در متغیر های کلاس ذخیره می کنیم به علاوه به تعداد فرزندان آرایه ایی دینامیک از جنس Human** ایجاد می کنیم

در default constructor از constructor قبلی استفاده می کنیم و مقادیر آن را برابر صفر قرار میدهیم

در copy constructor ازابتدا مقادیر ورودی را در شی قرار میدهیم سپس اشاره گر پدر،مادر و همسر ورودی و شی را برابر قرار می دهیم

و در نهایت با ایجاد آرایه دینامیک به طول تعداد فرزندان ورودی و از جنس Human** از double free جلوگیری می کنیم

Operator

Operator>: اگر سن شی بیشتر باشد true وگرنه false برمی گرداند

Operator==: در صورتی که تمام ویژگی های دو فرد به انضمام پدر و مادر و همسر یکی باشد true برمی گرداند

Operator+: ابتدا بررسی می کنیم که ایا دو غرد آیا همسر یک دیگر هستند سپس human * برای فرزند ایجاد می کنیم و نام خانوادگی آن را برابرنام خانوادگی پدر قرار میدهیم سپس کمک تابع srand هر یک از ویزگی های کودک را از پدر یا مادر او می گیریم .حال تعداد فرزندان پدر و مادر را یکی افزایش داده و آرایه دینامیکی به طول آن ایجاد می کنیم و فرزندان قبلی به انضمام فرزند جدید ایجاد می کنیم و بعد از حذف آرایه قبلی فرزندان آرایه فرزندان را برابر آرایه دینامیک جدید قرار می دهیم سپس اشاره گر به آرایه جدید را برابر nullptr قرار می دهیم و در نهایت حذف می کنیم این کار را برای مادر نیز به همین ترتیب انجام می دهیم

در نهایت پدر و مادر فرزندان تعین می کنیم و اشاره گر به فرزند را بر می گردانیم

Operator++ : سن فرد مورد نظر را یکی اضافه می کند

isChildof : در صورتی که پدر شی با ورودی برابر باشد true بر می گرداند

IsFatherof :در صورتی که پدر ورودی شی باشد true برمی گرداند

isMotherOf : در صورتی که مادر ورودی شی باشد true برمی گرداند

printChidren : ابتدا ارایه ای دینامیک به طول تعداد فرزندان و از جنس string می نویسیم سپس با الگوی selection sort فزندان را مرتب کرده و در هر مرحله نام فردی که در مقایسه با دیگران کم ترین سن را دارد در آرایه دینامیک ایجاد شده ذخیره می کنیم و در نهایت از اخر ارایه دینامیک چاپ می کنیم چون در سوال حالت نزولی خواسته شده است و در نهایت آرایه دینامیک ایجاد شده را حذف می کنیم

Oracle

این کلاس در واقع شامل توابعی است که می خواهیم آن ها را روی human ایجرا کنیم و شی ایجاد شده از آن مفهوم خاصی جز نام فرد ندارد .ابتدا یک constructor می نویسیم که نام ورودی را برابر نام شی قرار دهد. حال به توضیح توابع می پردازیم

Marry :ابتدا شرایط ازدواج را برسی می کنیم اعم از سن بالای ۱۸ سال و تجرد سپس هر فرد را هممسر دیگری قرار میدهیم و در نهایی true برمی گردانیم در صورتی که شرایط ازدواج مهیا نباشد false بر می کردانیم

setChild : همانند الگوریتمی که برای +operator اعمال شد را انجام می دهیم با این تفاوت که فرزند در اینجا مشخص است و نیازی نیست فرزند را برگردانیم برای دسترسی به تعداد فرزندان از تابع getnumberofchildren استفاده می کنیم همچنین تابع دیگری با نام setNumberOfChildren نیز به کلاس human اضافه می کنیم تا بتوانیم به کمک آن تعداد فرزندان را نغیر دهیم
حال تعداد فرزندان پدر و مادر را یکی افزایش داده و آرایه دینامیکی به طول آن ایجاد می کنیم و فرزندان قبلی به انضمام فرزند جدید ایجاد می کنیم و بعد از حذف آرایه قبلی فرزندان آرایه فرزندان را برابر آرایه دینامیک جدید قرار می دهیم سپس اشاره گر به آرایه جدید را برابر nullptr قرار می دهیم و در نهایت حذف می کنیم این کار را برای مادر نیز به همین ترتیب انجام می دهیم

isFamily :میدانیم که تمامی فامیل های سببی و نسبی را میتوان با کمک پدر ومادر و همسر و فرزندان تعین کرد به عنوان مثال برای برسی این که فردی پسر دایی فرد دیگری باشد باید پدر مادر فرد اول با پدر پدر فرد دوم برابر باشد یا آن که برای این که فردی نوه پدربزرگ پسر عمو دیگری باشد باید رابطه زیر برقرار باشد

H1->father ->father->(one of the children)->spouse->father = h2->father->father

همان طور که مشاهده می شود ۵ حالت وجود دارد که اصطلاحا یک میخ زده شود پدر،مادر،همسر،یکی از فرزندان و این که نیاز به رابطه نباشد این به این معنی است که به عنوان به مثال در مثال پسر دایی و با رابطه یا میخ مسله حل می شود و نباید رابطه اضافه تر زده شود

اما سوال انجاست که کدام یک از فرزندان انتخاب شوند؟

در این جا ما دو حلقه تو در تو داریم حلقه ای که رابطه بالا را تعیین می کند که این حلقه درون حلقه ایی دیگر است که تعیین می کند کدام یک از فرزندان وارد رابطه شوند

اما ایده اصلی چیست؟

گفته شد که برای هر بخش از یک رابطه ۵ حالت وجود دارد پدر،مادر،همسر،یکی از فرزندان و این که نیاز به رابطه نباشد به علاوه ۴ تا از این رابطه ها را نیز در نطر می گیریم میشود با افزایش این عدد به فامیل های خیلی دور تر نیز میتون رسید تا زمانی که کل کره زمین را با استفاد از پدر اول آدم تعیین کرد که نیاز سوال نیست

بنابراین ما ۵^۴ رابطه داریم که هر یک مشخص کننده هر یک از افراد فامیل است که ممکن است برخی از این رابطه ها با هم هم ارز باشند که در این بخش نیاز به حل آن نیست و در بخش getFamily به این موضوع می پردازیم

اما ۵^۴ حالت تمام حالات نیست چون ما نمی دانیم کدام یک از فرزندان باید انتخاب شوند بنابراین numberOfChildren^4 * 4^5 حالت وجود دارد

توابع کمکی Base و Base_numberOfChildren نیز برای تبدیل عدد شمارنده به عددی در مبنای به ترتیب ۵ و تعداد فرزندان می باشد

حال ابتدا حلقه ای با شمارنده ای به اندازه numberOfChildren ^4 برای تعین این که کدام یک از فرزندان در هر بخش از رابطه دخیل بشوند مینویسم و با کمک تابع و Base_numberOfChildren آن را به مبنای تعداد فرزندن میبریم سپس حلقه ایی درون این حلقه تعریف می کنیم که شمارنده آن به طول ۵^۴ است ایجاد می کنیم سپس با کمک تابع Base شمارنده را به مبنای ۵ می بریم در مثلا فرض کنیم تعداد فرزندان ۳ باشد و عدد شمارنده اول ۷ و عدد شمارنده دوم ۸۹ باشد که به ترتیب (۰۰۲۱) و (۰۳۲۴) که را بطه حاصل به صورت زیر می باشد

T1 ->spouse ->mother->children[1]

به همین ترتیب تولید این رابطه باید برای فرد ورودی دوم نیز انجام شود که منجر به ۴ حلقه تو در تو میشود

برای آنکه هر یک از اعداد تولید شده در مبنا های داده شده را تبدیل به المان های رابطه کنیم باید از ۴ ارایه هر یک برای هر حلقه استفاده می کنیم که به عنوان مثال اگر arr1[0] =2 اولین المان فرد اول mother می شود

به این ترتیب رابطه فامیلی را بررسی کرده و در صورت فامیل بودن true بر می گردانیم

getFamily:

همانند تابع isFamily عمل می کنیم با این تفاوت که باید خروجی تکراری حاصل از روابط را به فامیل اضافه نکنیم برای این کار یک آرایه دینامیک از جنس human** و نام family ایجاد میکنیم و طبق الگوریتم getfamily در صورتی که نتیجه یک رابطه nullptr نشود آن را به family اضافه می کنیم اما قبل از اضافه کردن باید بررسی کنیم که ایا فرد داده شده قبلا اضافه شده یا خیر برای این کار با استفاده از یک حلقه و بررسی اینکه ایا فرد داده شده جزیی از فامیلی هست یا خیر اضافه کردن آن را تعیین می کنیم که اگر چنین فردی نباشد متغیری که برای شمارش اعضای فامیل به نام numberOfFamily را اضافه می کنیم و در نهایت آرایه family را برمیگردانیم

getPopulationOfFamily: برای این کار تابع getFamily را روی ورودی صدا می زنیم و در با کمک یکwhile تعداد آن را می شماریم و این تعداد را بر می گردانیم

تذکر:دقت شود برای امتحان کردن ۳ تابع پایانی اطلاعات تمام فرزندان داده شود در صورتی که متغیری تعریف شود که تعداد فرزندان آن ۳ باشد و فرزندان آن مشخص نشده باشند بررسی خانواده آن ممکن نیست

Data Base

روند حل سوال به این صورت آغاز می شود که ابتدا جداول خواسته شده را در فایل insert.sql ایجاد میکنیم برای این کار باید primariy key و foreign key را تعیین کنیم .میدانیم primary keyبه عنصری یا عناصری اختصاص می یابد که به طور خاص عناصر جدول را مشخص می کند و foreign key تعیین کننده عناصری است که primary key از جدول دیگر است و به این ترتیب در صورتی که چنین primary key در جدول reference نباشد اجازه insert را نمی دهد حال primary key و foreign key هر یک از جداول را تعیین می کنیم.

Users: primary key: id foreign key No

Blockuser: primary key: (blocker_user_id ,bocked_user_id)

foreign key: foreign key(blocker_user_id) references users(id),

foreign key(blocked_user_id) references users(id)

Message: primary key: id

Foreign key: foreign key(sender_id) references users(id),

foreign key(receiver_id) references users(id)

Channel: primary key : id

Foreign key: foreign key(creator_id) references users(id)

Group: primary key: id

Foreign key: foreign key(creator_id) references users(id)

GroupMessage: primary key: id

Foreign key: foreign key(sender_id) references users(id),

foreign key(group_id) references groups(id)

ChannelMessage: primary key: id

Foreign key: foreign key(channel_id) references channel(id)

MessageAttachment: primary key: message_id

Foreign key: foreign key (message_id) references Message(id)

GroupMessageAttachment: primary key: message_id

Foreign key: foreign key (message_id) references GroupMessage(id)

ChannelMessageAttachment: primary key: message_id

Foreign key: foreign key (message_id) references ChannelMessage(id)

ChannelMembership: primary key: primary key(user_id ,channel_id)

Foreign key: foreign key(user_id) references users(id),

foreign key(channel_id) references channel(id)

GroupMembership: primary key: primary key(user_id ,group_id)

Foreign key: foreign key(user_id) references users(id),

foreign key(goup_id) references groups(id)

حال بر حسب نیازی که هر سوال دارد insert می کنیم و در فایل insert.sql ذخیره می کنیم

Query های خواسته شده را در فایل query.sql ذخیره می کنیم حال توضیح مختصری برای هر بخش می دهیم

۱.با کمک دستور users اطلاعات فرد خواسته شده را select می کنیم

۲.با کمک دستور update ایمیل فرد را از users تغییر می دهیم

۳.کانال هایی که فرد عضو هست را از channelmembership پیدا کرده و عنوان آن را select می کنیم

۴.ابتدا کانال خواسته شده را با کمک id آن پیدا کرده سپس دستور count تعداد اعضای آن را از channelmemberdhip می شماریم

۵.با کمک دستور like این کار را می کنیم علامت ٪ به این معنی است که هر چیزی بعد از ۰۹۳۵ می تواند باشد

۶.از users id افرادی که توسط فرد مذکور block شده اند را می یابیم و با کمک دستور and بررسی می کنیم که زمان blockیک ماه اخیر باشد

۷.ابتدا با دستور group by کانال هایی که بشتر از ۳ عضو دارند را می یابیم سپس با کمک and بررسی می کنیم که ایا فرد مذکور عضو کانال هست و در نهایت کانال هایی را که این ویژگی ها را دارد را از users شماره تلفن صاحبان آن را بر می گردانیم

۸.با دستور and بین دو دستور این کار را می کنیم دستور اول آن که ۱۰ پیام آخر از message باشدو دستور دوم این که بین دو طرف مذکور رد وبدل شده باشد که دو حالت دارد یا فرستنده طرف اول و گیرنده طرف دوم است ویا بلعکس

در بخش دوم سوال با کمک دستور delete تمام پیام هایی که فرستنده طرف اول و گیرنده طرف دوم است ویا بلعکس را حذف می کنیم.

۹.برای این کار ابتدا یک join می زنیم روی channel_id و creat_at از channelmessage سپس کانال هایی که دریک ماه اخیر پیامی نداشته اند را با کمک distance انجام داده و ادامه را همانند خواسته مسله انجام میدهیم

بخش چهارم

برای آن که چند adminداشته باشیم یک attribute از جنس ارایه تعریف می کنیم ونام کانال جدید را channel_morthanAdmin می گذاریم به همین نحو برای گروه عمل کرده و نام آن را group_morethanAdmin می گذاریم این دو جدول نیز در پایان فایل creat.sql قرار داده شده است.

Insert این دو جدول نیز در انتهای فایل insert.sql قرار داده شده است نکته قابل اهمیت باید برایinsert ارایه همانند admin_id به طریق زیر عمل کنیم مثلا ’{ {100200},{123456} }‘