

به نام خدا

سوال 1:

الگوریتم و روند حل سوال کاملاً در فایل همراه صورت سوالات، تشریح شده بود. توابع هم همان توابع هستند. فقط چند نکته:

1. در تابع `getHeight` با توجه به اینکه در ساختار درختی در هر طبقه به تعداد

توان های 2 عدد داریم، میتوان تعداد کل اعداد موجود در آرایه یا اندازه آن را به

کمک مجموع توانهای 2 از 2^0 و ... نوشت. (تعداد این توانها + 1) تعداد طبقات است. که چون توانها را از صفر در نظر گرفتیم ارتفاع را به ما خواهد داد.

2. در اپراتور `<<` که برای چاپ کردن استفاده شده از همین ارتفاع و تعداد طبقات

بهره گرفتیم. این تابع جزو توابع کلاس نیست و برای مرتب شدن کد در فایل کد

C++ کلاس نوشته شده است.

3. تابع `Swap` هم جزو توابع کلاس نیست و به همین دلیل هم در کنار کد کلاس

قرار گرفته.

سوال 2:

این سوال هم طبق چیزی که در صورت توضیح داده شده دارای یک پوینتر است

که مقدار حافظه ای که برای داده ها در نظر گرفتیم Capacity نامیدیم و تعداد اعدادی که در وکتور جای گرفته Size نامیدیم. اندازه وکتورها با توجه به پارامتر دوم مقایسه شده.

در مورد Copy constructor باید گفت که چون متغیر دینامیکی داریم که باید Delete هم بشود، اگر فقط ادرس پوینترها را مساوی قرار دهیم دوبار پاک خواهیم کرد که سبب ایجاد اشکال و ایراد است. بنابراین متغیر دینامیکی جدید می سازیم، برایش خانه حافظه اختصاص می دهیم و عناصر را کپی می کنیم. در مورد Move constructor وقتی یک Rvalue داریم به جای اینکه یک جای جدید از حافظه بگیریم و ... پوینتر جدید را مساوی پوینتر چیزی که می خواهیم از رو آن کپی کنیم قرار می دهیم و پوینتر قبلی را nullptr قرار می دهیم که دیگه به آن نقطه اشاره نکند و با از بین رفتنش اطلاعات ما را هم از بین نبرد و دوبار پاک کردن یک نقطه اتفاق نیفتد. تذکر: در نظر گرفته ام که همواره اندازه باید کمتر از نصف ظرفیت باشد.

سوال 3:

این سوال طبق استاندارد بیان شده و اصلاح شده نوشته شده و کلیه توابع دو کلاس کار می کنند. فقط ذکر چند نکته در مورد نحوه نوشتن آن می تواند مفید باشد.

الف) یک پوینتر مثل پوینتری که قرار است به فرزندان اشاره کند تعریف شده به اسم family که به هرکسی که با این فرد فامیل باشد اشاره خواهد کرد. به این نحو که یک تابع setFamily تعریف شده که هرگاه قرار است نسبتی بین افراد ایجاد شود فراخوانی می شود. هم برای دو یا چند فرد، هم برای همه فامیل های آنان. یعنی این فرد علاوه بر خود فرد با تمام فامیل های طرف مقابل هم فامیل می شود. برای نوشتن توابع Oracle هم از همین تابع و پوینتر بهره گرفته شده.

ب) فقط هنگام نسبت دادن چیزی به پوینتر های دو بعدی New شده اند. تا از اشغال بی جهت حافظه خودداری شود. پوینتر های موجود در پوینتر دو بعدی هم برایشان اشغال حافظه و New اتفاق نیفتاده و فقط آدرس انسانها را در خود نگه می دارند تا از Double free و اشغال بیهوده حافظه جلوگیری شود.

ج) در اپراتور + برای نگه داشتن فرزند New اتفاق افتاده و آدرس پوینتر آن بر می گردد که باید حتما delete شود. چون آدرس به main بر می گردد پس مجبوریم در Main آن را delete کنیم.

آپلود :

یک Repository جدید ساخته ، با کمک دستورات در command line

ارسال می کنیم.

لینک:

<https://github.com/k-ghodsifar/AP-HW3.git>