

Hw3

9623071

C++

سوال ۱:

هدف کلی سوال ساخت درخت باینری میباشد
بصورتی که گره پدر بزرگترین مقدار را دارد
و گره های فرزند کمتر یا مساوی گره پدر را دارند
در حل این سوال از توابعی استفاده شده که در کد توضیح
آنها داده شده است

همچنین برای توضیح بیشتر

Add(key)

تابعی که یک مقدار را به درخت اضافه میکند

Delete()

تابعی که آخرین مقدار درخت را حذف میکند

Max()

تابعی که بزرگترین مقدار درخت که همان پدر اصلی است را

میدهد

getHeight()

تابعی که ارتفاع درخت را باز میگرداند

Parent(i), LeftChild(i), RightChild(i)

توابعی که به ازای کلیدی که به عنوان ورودی میگیرند

مقدار پدر یا فرزند چپ یا راست را بر میگردانند

printArray()

تابعی که درخت را به صورت ردیفی و درکنار هم چاپ
میکند

Operator[i]

تابعی که به ازای کلیدی که به عنوان ورودی میگیرد مقدار
آن را باز میگرداند

همچنین این تابع ساختار درخت را مرتب میکند

به دلیل استفاده از اپراتور های $+$ $-$ $=$ که ممکن است
مقدار یک گره را عوض کنند

Operator<<

همچنین تابعی که درخت را مانند درخت چاپ میکند

heapsort()

تابعی که مقدار درخت را از بزرگ به کوچک چاپ میکند

توابع کمکی نیز مورد استفاده قرار گرفته شدند

`max_heapify(array, sizeof array, indx)`

تابعی که اگر مقدار گره پدر از فرزندش کوچکتر بود پدر و
فرزند را جابه جا میکند

`Build_maxheap(array, sizeof array)`

تابعی که آرایه به کمک تابع قبل به صورت درخت میسازد

`Par(i), Left(i), Right(i)`

توابعی که به ازای کلید ورودی کلید پدر یا فرزند چپ و
راست را برمیگردانند

همچنین این کلاس سه مدل constructor دارد

که با لیست و آرایه و بدون ورودی کار میکنند

همچنین copy constructor کلاس نیز تعریف شده است
در destructor کلاس حافظه ی گرفته شده جهت ذخیره سازی
درخت به سیستم بازگردانده میشود
پینوشت:

اپراتور های + و - باز نویسی نشدند اما خواسته ی سوال
برآورده میشود

سوال ۲

هدف سوال پیاده سازی کلاس وکتور با متغیر های integer
میباشد

در کد توضیح توابع داده شده است
برای توضیح بیشتر:

`push_back(key)`

تابعی که مقداری را به انتهای وکتور اضافه میکند

`pop_back()`

تابعی که مقدار انتهای وکتور را پاک میکند

`<, ==`

اپراتورهای مقایسه گر

این اپراتور ها طول وکتور را ها را مقایسه میکنند

`Length()`

طول وکتور ها برابر است با مجذور مجموع توان دو های

المان های وکتور

Copy constructor

تمام مقادیر یک وکتور ورودی را برای وکتور کپی میکند

همچنین وکتور ورودی lvalue میباشد

Move constructor

تمام مقادیر وکتور ورودی را برای وکتور کپی میکند

همچنین وکتور ورودی rvalue میباشد

که تابع make_vector برای مجبور کردن کامپایلر های

هوشمند تر برای استفاده از move constructor میباشد

در destructor کلاس حافظه ی آرایه ی گرفته شده به سیستم

بازگردانده میشود

همچنین توابعی که مقدار size , capacity وکتور را

برمیگردانند

همینطور سه مدل اپراتور = برای کلاس تعریف شده

یکی وکتور lvalue یکی وکتور lvalue یکی list را برابر
وکتور قرار میدهند

سوال ۳

با توجه به اینکه در صورت سوال توضیحات در مورد توابع
داده شد و فقط پیاده سازی باید انجام میشد توضیحات در
کد میباشد

همچنین توابع زیر مربوط به class oracle نوشته نشدند

IsFamily(), getFamily(), getpopulationOfFmily()