

صف اولویت دار!

پی دی اف سوال در تمرین آپلود شده است.

دوراهی! (binary tree)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

سپهر ک مدتی ست مسیر خود را گم کرده است به پیشنهاد دوستش، به نزد فردی رفته تا به کمک او بتواند مسیر خوب از بد را تشخیص داده فرد به سپهر پیشنهاد داد در انتخاب مسیبردست بین دوراهی ها و رسیدن به چیزی که میخواهد از ساختاری شبیه درخت دودویی استفاده کند (: به طوریکه وقتی از نقطه ای شروع میکند نقطه با انرژی کمتر(بد) را در یک سمت و نقطه با انرژی بیشتر(خوب) را در سمت دیگر قرار دهد تا وقتی میخواهد به یک نقطه انرژی خاص الهی برسد بتواند سریع صراط مستقیم را پیدا کند! LOL به سپهر کمک کنید تا این کلاس را طراحی کند.

هدف : پیاده سازی یک کلاس درخت دودویی همراه با استفاده از لیست پیوندی
[لینک توضیح درخت دودویی](#)

توابع مورد نیاز:

- destructor و constructor توابع
- تابع copy constructor
- تابع افزودن یک گره به درخت یک عدد صحیح گرفته و طبق قوانین درخت به درخت اضافه می شود.
- تابع جست و جو: یک عدد صحیح را گرفته و در صورت وجود آن گره را بر میگردداند در غیر این صورت مقدار نال برگشت داده شود.
- تابع چاپ درخت در این تابع میخواهیم اعضای درخت را با ترتیب خاصی که با آن پیمایش preorder گفته می شود نمایش دهیم. [لینک توضیح پیمایش preorder](#)

نکته ۱: حتما این تابع بصورت بازگشتی طراحی شود. نکته ۲: حتما کد خود را بصورت چند فایل مجزا بنویسید. قسمت امتیازی:

- پیاده سازی توابعی برای نمایش پیمایش inorder و postorder درخت دودویی

- پیاده سازی تابع delete یک عدد صحیح از کاربر گرفته و آن گره را از درخت حذف کنید.

لیست دو بعدی (operator overloading)

یک کلاس به نام List برای ذخیره سازی تعداد نامحدود داده که از هر نوعی می‌توانند باشند ایجاد کنید.

این کلاس علاوه بر متد های مورد نیاز باید شامل متدهای زیر نیز باشد:

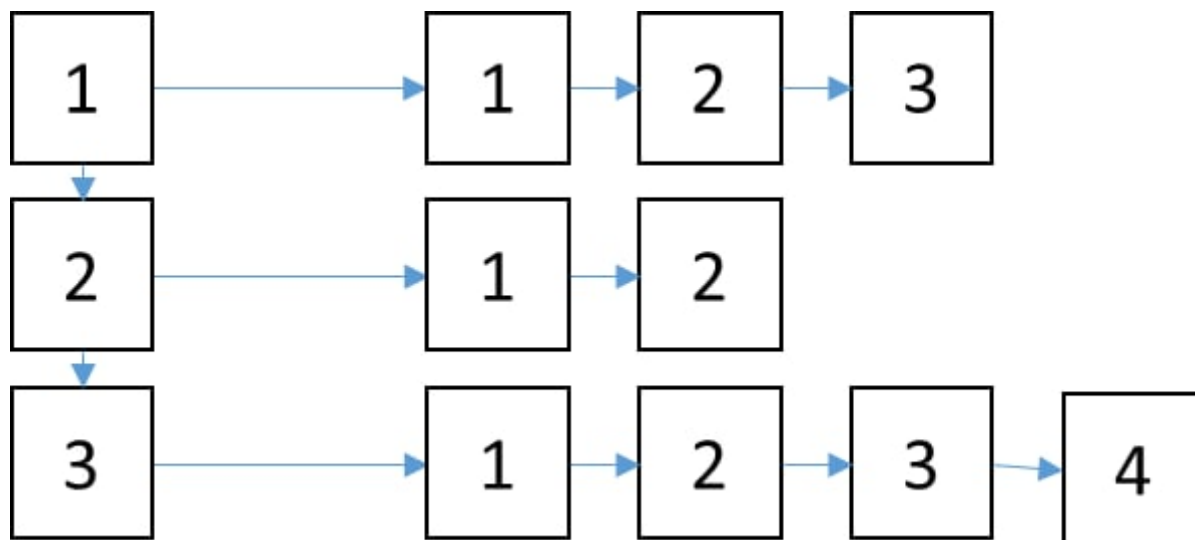
- تابع remove که یک مقدار را از آخر لیست حذف کند.
- تابع insert که یک مقدار را به آخر لیست اضافه می‌کند.
- (امتیازی) عملگر [] که با استفاده از آن بتوان به عناصر لیست دست پیدا کرد و مقدار آن را تغییر داد.

در این قسمت می‌خواهیم با استفاده از کلاس List در قسمت قبل کلاسی به نام List2D طراحی کنیم که با استفاده از آن بتوانیم لیست های دو بعدی ایجاد و آنها را مدیریت کنیم.

این کلاس علاوه بر متد های مورد نیاز شامل متدهای زیر است:

- تابع insert که بعنوان ورودی یک لیست می‌گیرد و آن را به آخر اضافه کند.
- تابعی دیگر به همان نام insert که یک داده از همان جنس List می‌گیرد و یک لیست به آخرین لیست ها اضافه می‌کند که در آن تنها این عضو وجود دارد. (برای مثال در صورتی که تمامی مقادیر از نوع int باشند، insert(3) یک لیست به آخر لیست ها اضافه می‌شود که تنها شامل عدد ۳ است.
- (امتیازی) اپراتور () که یک int به عنوان ورودی می‌گیرد و لیست متناظر با index این عدد را برمی‌گرداند.
- استفاده کردن از عملگر () به صورت (ستون، ردیف) برای نشان دادن داده موجود در آن خانه و امکان ویرایش داشته باشد.
- با استفاده از عملگر <> می‌توانیم عناصر لیست را در خروجی استاندارد چاپ کنیم. به طوریکه عناصر هر لیست در یک خط چاپ شوند.
- (امتیازی) اپراتور = که شیئی اول را در شیئی دوم بریزد. (توجه کنید این دو شیئی حافظه های هم مستقل داشته باشند).

شکل زیر یک نمایشی از لیست ها در کلاس List2D است:



برای مثال اگر شکل بالا محتوای اوجکت lst را نشان می‌دهد. lst(1) باید لیست اول را برگرداند و همچنین lst(1,2) باید مقدار عنصر دوم از لیست شماره 1 را برگرداند.

موارد مهم:

- توجه داشته باشید کلاس شما باید بتواند با اوجکت های از نوع const هم کار کند.
- هر متد دیگری که این کلاس می‌تواند نیاز داشته باشد را اضافه کنید. (برای مثلا ستر و گتر و...)
- توجه داشته باشید رعایت اصول شی‌گرایی و encapsulation الزامی است.
- طراحی کلاس های شما باید در چند فایل در صورت نیاز باشد.