

توضیحات تمرینات ساختمان داده و الگوریتم

سارا مظاهری

سوال L1:

در این سوال قرار است با لیست پیوندی که یکی از پیاده سازی اعداد است، عددهای صحیح و خیلی زیاد را باهم جمع کنیم. مثلاً اعداد 109+ یا 456- را به صورت دو لیست پیاده سازی کنیم. (در لیست باید از + و - ، همچنین اعداد 0 تا 9 نیز استفاده شود. طبق سوال لیست ها سرلیست دارند که علامت اعداد نیز در خود ذخیره میکنند.

ایده و توضیحات سوال L1

ابتدا یک کلاس LINKED LIST تعریف کردم که در آن تابع های مختلف با صفات گوناگون در آن جای داشت. ابتدا به توضیح تک تک توابع میپردازیم.

- تابع ADD FIRST: در این تابع قرار است یک NODE جدید در ابتدای لیست پیوندی تو ایجاد کنیم، پس head را به X میدهیم، سپس NODE را وارد لیست پیوندی میکنیم.
- تابع ADD LAST: در این تابع temp را به عنوان head قرار میدهیم و هنگامی که NIL نبود به آخر لیست اضافه خواهیم کرد.
- تابع REMOVE FIRST: همان طور که از اسمش پیداست قرار است این تابع حذف کند.
- تابع ADD TWO LISTS: مهم ترین تابع در این کلاس میباشد. زیرا قرار است دو تابع را باهم جمع کند. در این تابع دو صفت FIRST و SECOND داریم که به ترتیب یعنی اولین رقمی که قرار بگیریم و دومین رقم که باهم جمع شود. اگر هر عدد را در لیست بخواهیم قرار دهیم از راست به چپ لیست دارای یکان ، دهگان ، صدگان الی آخر است. اگر یکان دو عدد NIL نبود، باهم جمع میشوند و در یک متغیر ریخته میشوند sum_value. حال اگر جمع یکان ها بزرگتر از 10 شد، دهگان در متغیری به نام carry نگه داری میشود و بعد به دهگان اضافه میشود. (یا به عبارتی دیگر باقیمانده عدد 10 که در یکان به دست آمد میشود یکان جمع دو لیست، و خارج قسمتش میشود متغیر carry)
- تابع PRINT LISTS: قرار است اعدادی که در data ذخیره شد را چاپ کند.

* برای تفریق کردن ایده جالبی را که انجام دادم را بگویم:

تفريق کردن دو لیست از همدیگر کار بسیار پیچیده ای میشد، زیرا اگر به عنوان مثال یکان عدد اول کوچکتر از یکان عدد دوم بود آنگاه باید از NODE قبلیش استفاده میشد، دوباره به NODE بعدی اضافه میشد و این کار خیلی پیچیده اش میکرد و از طرفی اگر جواب تفريق منفی میشد دوباره برای ثبت منفی به سرلیست احتیاج داشتیم که باز قضیه دشوار تر میشد. پس ایده ای به ذهنم رسید که با استفاده از جمع، تفريق نیز انجام شود. به عنوان مثال فرض کنید بخواهیم عدد 1492 را منهای عدد 1066 کنیم. چون یکان عدد اول کوچکتر است، پس باید از قبلی بگیریم که کار دشواری است. پس آمدم گفتم از تمام NODE های 9 تا کم میکنیم به غیر از یکان که باید 10 تا ازش کم کنیم. یعنی (10-6), (9-6), (9-0), (9-1). حال عدد دوم را به علاوه عددی که به دست آمد یعنی 8934 میکنیم. $1492+8934=10426$ رقم سمت چپ را حذف کرده و جواب به دست می آید یعنی 426: (میتوانید خودتون یکبار حساب کنید تا ببینید منها کردن با استفاده از جمع خیلی آسون میباشد. حال مشکلی که پیش اومد این بود که اگر عدد اولمون کوچکتر از عدد دوممون بود چه کاری انجام دهیم؟

اومدم گفتم اگر عدد اول کوچکتر از عدد دوم بود، عدد دوم را کم میکنم و کنارش یک منفی قرار میدهم یعنی اینکه مثلاً 5-3 میشود $3-5$ که کنارش منفی میگذاریم و عدد 2- به دست می آید.

سوال L2:

در این سوال از ما میخواهد با یکی از ساختار داده ها چند جمله ای که ضریب C دارد و ضریب توان های x,y,z هستند اعمال زیر را انجام دهیم

- چاپ یک عبارت
- تعیین بیشترین عمق
- کپی کردن یک عبارت
- جمع و تفریق دو عبارت
- مشتق گیری یک عبارت

ایده و توضیحات سوال L3:

اگر یک چند جمله ای را متصور شویم که درجه و ضریب های مختلف دارد مانند:
 $p1(x) = 8x^3 + 3x^2 + 2x + 6$ در نتیجه میتوان آن را در یک NODE قرار داد به طوریکه متغیر اول در head ضریبش و در temp درجه توان نوشته میشود. برای مثال بالا این NODE را در نظر بگیرید:

8	3	->
---	---	----

3	2	->
---	---	----

2	1	->
---	---	----

6	0	NULL
---	---	------

حال با رسم این NODE ها میتوان به راحتی جمع و یا تفریق عبارات را حساب کرد. به طوری که اگر عملگرمان جمع بود اگر توان هایی یکسان داشت آنها را جمع میکنیم و اگر نداشت تنها کنار هم میگذاریم بدین سان تفریق را هم به دست میآوریم.

برای تعیین بیشترین عمق یا همان بیشترین درجه هر یک از عبارات چندجمله ای کافیت ماکزیمم temp را به دست بیاریم یعنی در هر کدام از NODE ها بزرگترین را پیدا میکنیم و چاپ میکنیم. و در آخر مشتق گیری از آن مشتق گیری جزیی میباشد، یعنی باید از درجه توان یکی کم کنیم، سپس ضرب در ضریب آن بکنیم و چند جمله ای را چاپ کنیم. بدین منظور در مثال بالا در اولین NODE یکی از temp کم میکنیم و به head را همان ضریبمان اضافه میکنیم تا آخرین NODE. اگر درجه صفر بود باید کل NODE را از بین ببریم یا همان NULL کنیم.

برای کپی کردن یک چند جمله ای، فکر میکنم منظور سوال همان چاپ کردن باشد. پس به آسانی لیستمان را چاپ میکنیم.

سوال L3:

در این سوال نیاز به الگوریتمی داریم که عبارت میان وندی را به عبارت پسوندی و پیشوندی تبدیل کنیم. طبق صورت سوال باید با استفاده از ساختار داده پشته استفاده کرد.

ایده و توضیحات سوال L3

ساختار داده پشته به این صورت است که آخری که وارد میشه، اولی ای هست که نیز خارج میشود. در واقع سیاست پشته last in- first out میباشد. توابعی که باید در کلاس stack استفاده کنیم عبارتند از موارد فوق میباشد:

- تابع PUSH: این تابع یک عنصر را در stack بالایش درج میکند.
- تابع POP: این تابع خارج و حذف میکند از بالای پشته یا stack.
- تابع IS EMPTY: مشخص میکند که آیا پشته خالی است یا خیر
- و دیگر توابع که در کد L3+ استفاده شده...

*برای این سوال ایده بهتر و آسون تری به ذهنم زد که قدم را کوتاه تر و جالب تر کرد:

با استفاده از الگوریتم preorder، inorder و postorder که در ذیل تعریف کردیم یک کلاس داریم به نام B Tree و یک کلاس داریم TreeNode که NODEهای درخت هستند. یک عبارت میدیم و بعدش پشته تعریف میکنیم. حال برای هر کارکتر توی اون عبارت اگر غیر از (و) بود یک NODE درست میکنیم و آن NODE را داخل پشته قرار میدهیم مانند الگوریتم که گفته میشود. در ادامه اگر جایی از عبارت (را دیدیم باید 3 چیز را از بالای پشته pop کنیم و بعد کانکشنی ایجاد میکنیم و بعد میگذاریم داخل stack و در آخر چاپ میکنیم...

Algorithm buildExpression(E):

Initialize a new empty stack for holding binary expression trees.

For each character of the input string E:

If the character is a number or operator x,

- a. Create a single-node binary tree T whose stores x
- b. Push T onto the stack

On the other hand, if the character is a close-parent””)”

- a. Pop the top three trees from the stack (these represent a subexpression: E1 o E2)
- b. Attach the trees for E1 and E2 as the children for o
- c. Push the resulting tree back onto the stack

نمادگذاری میان وندی، پس وندی، پیش وندی در صورت ذیل است:

نماد گذاری پسوندی	نمادگذاری پیشوندی	نمادگذاری میان وندی
AB+	+AB	A+B
AB+C*	*+ABC	(A+B)*C

