

ساختمان داده - دکتر آبین

امیرحسین منصوری - ۹۹۲۴۳۰۶۹ - تمرین سری ۴

توضیح: کدهای مربوط به هر سوال در فولدر Solutions به صورت فایل‌های جداگانه قرار دارند. همچنین ساختمان داده‌های استفاده شده و توابع مربوط به آن‌ها در فایل‌های DS4_LinkedList.seudo و DS4_DoublyLinkedList.pseudo قرار دارند.

سوال ۱ - الف) (فایل DS4_1.pseudo)

توضیح: ابتدا به ترتیب نیمه اول لیست پیوندی داده شده را به یک لیست پیوندی جدید اضافه می‌کنیم. سپس با شروع از نیمه دوم لیست پیوندی داده شده، در هر مرحله با گرفتن عنصر آخر این لیست پیوندی و حذف آن، این عنصر را با عنصر بعدی لیست پیوندی داده شده مقایسه می‌کنیم. در صورت مغایرت، لیست داده شده متقارن نیست.

سوال ۱ - ب) (فایل DS4_1.pseudo)

توضیح: مشابه سوال ۱-الف عمل می‌کنیم، با این تفاوت که این بار در هر مرحله عنصر اول لیست پیوندی ساخته شده را می‌گیریم و مقایسه را انجام می‌دهیم.

سوال ۲ (فایل DS4_2.pseudo)

در صورتی که لیست داده شده طول ۱ داشته باشد، معکوس لیست برابر با خودش است و بنابراین به عنوان شرط اولیه این تابع بازگشتی استفاده می‌شود. در نهایت برای لیست‌های با طول بزرگتر از یک، عنصر اول لیست را حذف می‌کنیم، لیست باقی‌مانده را به صورت بازگشتی معکوس می‌کنیم، و عنصری که حذف کردیم را به آخر لیست اضافه می‌کنیم.

سوال ۳ - الف)

لیست پیوندی بهترین گزینه است. زیرا علاوه بر مصرف حافظه بهینه، این ویژگی را دارد که می‌توان در انتهای انجام مقایسه، عناصر باقی‌مانده لیست را با Loading... عمل و به صورت ساده‌ای به انتهای لیست نهایی اضافه کرد و نیازی به کپی کردن تک‌تک عناصر باقی‌مانده نیست.

سوال ۳ - ب) (فایل DS4_3.pseudo)

کافی است که از عنصر اول هر لیست شروع کنیم، و در هر مرحله با مقایسه کردن عضو فعلی دو لیست، عنصر کوچکتر بین دو لیست را در لیست نهایی درج کنیم. در نهایت عناصر باقی‌مانده احتمالی را با یک حرکت به آخر لیست نهایی اضافه می‌کنیم.

سوال ۴ - الف)

لیست پیوندی دو طرفه بهترین گزینه است. زیرا اجازه جابه‌جایی دو طرفه بین عناصر را (که در حل مسئله مورد نیاز است) به راحتی می‌دهد. همچنین حذف و درج عنصر در میانه لیست در آن بهینه است.

سوال ۴ - ب) (فایل DS4_4.pseudo)

ابتدا یک لیست پیوندی دو طرفه می‌سازیم که در آن حروف کلمه رمزگشایی شده ذخیره می‌شود. سپس یک پوینتر به یکی از عضوهای این لیست ذخیره می‌کنیم که مکان فعلی cursor را نشان می‌دهد. همچنین یک عضو اضافی (با مقدار صفر) به ابتدای لیست اضافه می‌کنیم تا بتوانیم مکان cursor را که قبل از حرف اول است، ذخیره کنیم. در آخر کار این عضو اضافی حذف می‌شود. حال در هر مرحله، یک کاراکتر از فایل را که اطلاعات دکمه‌های کیبورد در آن ذخیره شده را می‌خوانیم. اگر کاراکتر خوانده شده ">" بود، در صورت امکان پوینتر را یک واحد به جلو جابه‌جا می‌کنیم. به طور مشابه اگر کاراکتر خوانده شده "<" بود، در صورت امکان پوینتر را یک واحد به عقب جابه‌جا می‌کنیم. همچنین اگر کاراکتر خوانده شده "-" بود، عضوی که پوینتر cursor به آن اشاره می‌کند را از لیست حذف می‌کنیم. اگر

کاراکتر خوانده شده کاراکتری غیر از موارد بالا بود، در مکانی که پوینتر cursor به آن اشاره می‌کند، حرف را به لیست اضافه می‌کنیم. به همین ترتیب کلمه رمزگشایی شده تشکیل می‌شود.

سوال ۵ (فایل DS4_5.pseudo)

ابتدا یک پوینتر به ابتدای لیست در نظر می‌گیریم. سپس پوینتر دیگری را که به همین عضو اشاره می‌کند را در نظر می‌گیریم. با جلو بردن پوینتر دوم، حاصل جمع عضوهایی که از آن‌ها عبور می‌کنیم را حساب می‌کنیم و همزمان آدرس این اعضا را در یک لیست دیگر ذخیره می‌کنیم. به محض این که حاصل جمع صفر شد، همه اعضای لیست ورودی که در لیست دومی ذخیره شده‌اند را از لیست ورودی پاک می‌کنیم. سپس تا آخر لیست جمع زدن را ادامه می‌دهیم. در ادامه پوینتر اول را جلو می‌بریم و پوینتر دوم را نیز مثل پوینتر اول مقدار می‌دهیم و همین کار را ادامه می‌دهیم تا پوینتر اول به انتهای لیست برسد.

سوال ۶ (فایل DS4_6.pseudo)

ابتدا یک پوینتر به ابتدای لیست در نظر می‌گیریم. سپس پوینتر دیگری را به عنصر بعدی‌ای که پوینتر اول به آن اشاره می‌کند، در نظر می‌گیریم. سپس با جلو بردن پوینتر دوم، هر عنصری که با عضوی که پوینتر اول به آن اشاره می‌کند برابر است را پیدا کرده و آن را حذف می‌کنیم. سپس پوینتر اول را یک واحد جلو می‌بریم و همین عمل را تکرار می‌کنیم، تا جایی که پوینتر اول به انتهای لیست برسد.