

۴: چرا Graph Data structure برای سبک های اجتماعی استفاده می شود؟
 به طور کلی دنیا استرکچرهای گرافی فرمت خاص به منقول سازمان حرفه و ذخیره سازی داده ها هستند که هوک در راستای نگهداری داده ها به منقول انجام پذیر از سبک های مد فکر در دنیا مربوط به طراری شده اثر قابتوان جاروشن های مناسب و از آن مهم ترین بهینه به دیتای مورد نظر در سترسی پیدا کرده و عملیات مربوط به را جابجایی الگوریتم های مختلف روی آن ها انجام در ادیتا شده آنچه ها به منقول شدن سازی مجموعه ای از اشیاء و ارتباطات مابین آن ها مورد استفاده قرار می گیرند و بر اساس نمودارهای (G, E, V)
 مناسب داده می شود.

I: چرا Dynamic Programming در حل مسائل پیچیده کاربرد دارد؟
 در حل برنامه نویسی پویا Dynamic Programming به چند دلیل در حل مسائل پیچیده کاربرد دارد برنامه نویسی پویا مسائل پیچیده را به زیر مسئله های کوچکتر و قابل مدیریت تر تقسیم می کند این زیر مسئله ها اغلب به صورت تکرار گشتی تعریف می شوند و ویژگی مهم این است که زیر مسئله ها با هم هم پوشانی دارند یعنی یک زیر مسئله ممکن است در حل چندین مسئله نیز تکرار شود ۲- ذخیره سازی نتایج زیر مسئله ها به جای محاسبه مجدد نتایج زیر مسئله ها به نام نویسی پویا این نتایج را ذخیره می کند معمولاً در یک جدول یا آرایه وقتی دوبار به یک زیر مسئله نیاز باشد نتیجه از حافظه خوانده می شود که به طور قابل توجهی زمان اجرا را کاهش می دهد این تکنیک ها با نام های Memoization یا جابجایی و Tabulation یا پیمین به نام شناخته می شود به نام نویسی پویا به دلیل توانایی در تقسیم مسائل به زیر مسئله ها ذخیره سازی نتایج جلوگیری از محاسبات تکراری و یافتن راه حل بهینه در حل مسائل پیچیده بسیار کاربرد دارد.

۱۴۰۴/۱۱/۲۵

Data structures and Algorithms بخش ۳

۴: HashTable چیست و چه کاربردی دارد؟ یک ساختار داده‌ای است که برای ذخیره و بازیابی سریع داده‌ها استفاده می‌شود. این ساختار داده‌ای بر اساس مفهوم کلید مقدار (key-value) کار می‌کند. به این معنی که هر داده (value) با یک کلید (key) مرتبط می‌شود. عملیات اصلی HashTable: ۱- دریافت کلید (key) زمانی که می‌خواهید یک داده را در HashTable ذخیره کنید یا به آن دسترسی پیدا کنید. ۲- بازیابی داده‌ها از زمانه ها کاربردی HashTable در بسیاری از زمینه‌ها کاربرد دارد از جمله پایگاه داده‌ها، برای اندیس گذاری و جستجوی سریع داده‌ها در پایگاه داده‌ها استفاده می‌شود. حافظه پنهان: برای ذخیره داده‌های پرکاربرد در حافظه پنهان استفاده می‌شود. قابلیت دسترسی به داده‌ها سریع‌تر می‌شود به طور خلاصه HashTable یک ابزار قدرتمند برای ذخیره و بازیابی سریع داده‌ها است که در بسیاری از زمینه‌ها کاربرد دارد. یاد رک مفاهیم اساسی و نحوه عملکرد آن می‌تواند از این ساختار داده‌ای به طور موثرتر در پروژه‌های خود استفاده کنید.

۵: B-TREE و BINARY TREE چه تفاوت‌هایی دارند؟ تفاوت‌های کلیدی بین دو درخت: ۱- تعداد فرزندان: درخت دودویی (BINARY TREE) حداکثر دو فرزند دارد. درخت B (B-TREE) می‌تواند تعداد زیادی فرزند داشته باشد. ۲- عمق: درخت B معمولاً از ۲ تا ۳ فرزند کم (مرتبه درخت B است ۲- ساختار گره: درخت دودویی هر گره معمولاً شامل یک مقدار داده و اشاره گرهایی به فرزندان چپ و راست است. درخت دودویی برای نمایش داده‌های سلسله مراتبی (مانند فایل سیستم و جداول) و مرتب سازی مانند درخت جستجوی دودویی پیاده سازی ساختارهای داده‌ای مانند هیپ. درخت دودویی ممکن است نامتوازن باشد و در بدترین حالت عملکردی مشابه لیست پیوندی داشته باشد.

A: Array و List چه تفاوتی دارند؟ تفاوت اصلی بین Array و List
 در برنامه نویسی به شرح زیر است: اندازه ثابت در مقابل اندازه پویا: Array (آرایه)
 اندازه آن در زمان تعریف مشخص می شود و پس از آن قابل تغییر نیست.
 List: اندازه آن پویا است و می تواند در طول اجرای برنامه تغییر کند. می توان
 عناصر را به لیست اضافه کرد یا از آن حذف کرد. نحوه ذخیره سازی: Array: عناصر
 آرایه به صورت متوالی در حافظه ذخیره می شوند این امر باعث دسترسی سریع
 به عناصر آرایه می شود زیرا آدرس هر عنصر به راحتی قابل محاسبه است.
 List: نحوه ذخیره سازی عناصر لیست به پیاده سازی آن بستگی دارد. عناصر
 در مکان های مختلف حافظه قرار می گیرند و به وسیله اشاره گرهای پیکه متصل
 می شوند.

B: Dictionary در Python چگونه کار می کند؟ Dictionary ها
 ساختارهای داده ای پرکاربرد هستند که به شما امکان می دهند
 داده ها را به صورت زوج کلید و مقدار ذخیره کنید و دسترسی به آن ها در Python
 ۱- غیر ترتیبی: دسترسی ها به صورت غیر ترتیبی هستند به این معنی که ترتیب درج عناصر
 حفظ نشده است. از Python 3.7 به بعد ترتیب درج حفظ می شود.
 ۲- قابل تغییر: شما می توانید به راحتی مقدار موجود در یک دسترسی را تغییر دهید یا
 آن را اضافه کنید یا حذف کنید. ۳- استفاده از کلیدهای گوناگون: هر دسترسی باید یک
 باشد و نمی توان دو مقدار با یک کلید گوناگون داشته باشید.

C: Tuple و List چه تفاوتی دارند؟ تفاوت اصلی بین Tuple و List
 در پایتون به شرح زیر است: تغییر پذیری (Mutability): List: قابل تغییر (Mutable)
 است یعنی می توان عناصر آن را بعد از ایجاد تغییر داد. اضافه کرد، حذف کرد، مقدار
 را عوض کرد. Tuple: غیر قابل تغییر (Immutable) است بعد از ایجاد نمی توان
 عناصر آن را تغییر داد. نحوه تعریف: List = [] با استفاده از براکت []
 تعریف می شود. Tuple = () با استفاده از پرانتز () تعریف می شود.

Date:

1KOK/1/20

Data structures and Algorithms

عاریزد:

List: برای ذخیره مجموعه ای از داده ها که ممکن است نیاز به تغییر داشته باشند.

Table: برای ذخیره مجموعه ای از داده ها که بنا بر تقییر کنند داشته
محتصات یک نقطه، اعدادات یک مورد به عنوان کلید در دیکشنری ها
dictionary استفاده می شود چون کلیدها باید immutable باشند.

D: Set در Python چرا برای حذف داده‌های تکراری استفاده می‌شود؟

در پایتون Set (مجموعه) یک نوع داده‌ای است که برای حذف داده‌های تکراری بسیار کارآمد است. دلیل این کارایی به ماهیت و نحوه عملکرد مجموعه‌ها برمی‌گردد.

۱- عدم اجازه به عناصر تکراری: مهم‌ترین ویژگی یک مجموعه این است که نمی‌تواند شامل عناصر تکراری باشد. وقتی شما یک لیست یا هر نوع داده تکراری را به یک مجموعه تبدیل می‌کنید، مجموعه به صورت خودکار تمام عناصر تکراری را حذف می‌کند و فقط عناصر یکتا را نگه می‌دارد. سادگی و خوانایی کد استفاده از مجموعه برای حذف تکراری‌ها بسیار ساده و خوانا است. مجموعه‌ها در پایتون به دلیل عدم اجازه به عناصر تکراری و پیاده‌سازی مبتنی بر Hash Table یک راه بسیار کارآمد و ساده و خوانا برای حذف داده‌های تکراری هستند.

E: Queue Stack چه تفاوتی دارند؟ تفاوت اصلی بین Queue و Stack (صف) در نحوه / اضافه و حذف عناصر است. Stack سیست: عملیات اصلی Push / اضافه کردن یک عنصر به بالای سیست. Pop حذف کردن عنصری از بالای سیست. کاربردها: مدیریت فراخوانی توابع در برنامه نویسی بررسی تعلق پیرانتزها در عبارات ریاضی پیاده سازی Undo/Redo

Recursion چیست و چرا در الگوریتم‌های پیچیده استفاده می‌شود؟
 یک تکنیک برنامه‌نویسی است که در آن یک تابع خودش را فراخوانی می‌کند تا یک مسئله را حل کند این فراخوانی‌ها کارهای ادغام پیدا می‌کند که به یک حالت پایه برسیم که مسئله به طور مستقیم قابل حل باشد در واقع هر فراخوانی تابع بازگشتی مسئله را به مسائل کوچک‌تر و ساده‌تر تقسیم می‌کند تا در نهایت به حالت پایه برسیم بازگشت یک ابزار قدرتمند در برنامه‌نویسی است که به ما امکان می‌دهد مسائل پیچیده را به روشی ساده و زیبا حل کنیم با این حال باید به ملاحظات مربوط به حالت پایه کارایی و محاسبات بازگشت توجه داشته باشیم بازگشت در بسیاری از الگوریتم‌های پیچیده و ساختارهای داده استفاده می‌شود.