1. Array و List چه تفاوتی دارند؟

تفاوت‌های اصلی بین Array و List (در زبان‌هایی مثل Java یا C# یا Python) در چند زمینه هست: ساختار، عملکرد، و کاربرد. اجازه بده به شکل ساده و دسته‌بندی شده توضیح بدم:

اندازه (Size)1.

Array (آرایه):

اندازه ثابت داره. وقتی ساخته شد، نمی‌تونی سایزش رو تغییر بدی.

مثال: int[] arr = new int[5];

List (لیست):

اندازه متغیر داره. می‌تونی بهش آیتم اضافه یا حذف کنی.

مثال در Java: List<Integer> list = new ArrayList<>();

نوع داده (Type)2.

Array:

معمولا فقط یک نوع داده خاص رو نگه می‌داره (در زبان‌های strongly-typed مثل Java یا C#).

List:

در اکثر زبان‌ها generic هست؛ یعنی می‌تونی مشخص کنی که چه نوع داده‌ای رو نگه داره.

مثلاً: List<String> یا List<Integer>

کارایی (Performance)3.

Array:

سریع‌تره در دسترسی به عناصر چون حافظه پیوسته داره.

List:

کمی کندتره چون ممکنه پشت صحنه resize بشه یا حافظه پراکنده داشته باشه (مخصوصاً در LinkedList ها).

امکانات (Features)4.

Array:  
ساده‌تره و امکانات کمتری داره.

List:  
امکانات بیشتر مثل add(), remove(), contains() و خیلی متدهای دیگه.

5. استفاده در زبان‌های مختلف

در Python:

list استفاده می‌شه به جای array.

ولی array واقعی هم با array module هست که کمتر استفاده می‌شه.

در Java:

int[] → آرایه

ArrayList<Integer> → لیست پویا

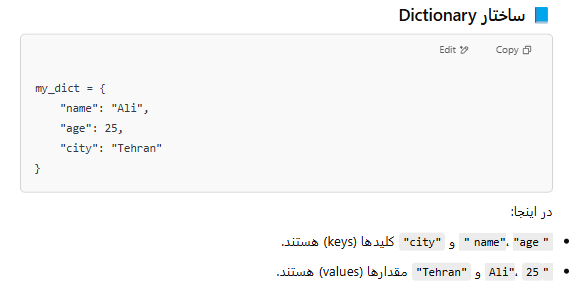
در C#:

int[] → آرایه

List<int> → لیست

1. Dictionary در Python چگونه کار می‌کند؟

در پایتون، (دیکشنری) یکی از انواع داده‌های پرکاربرد است که برای ذخیره‌سازی مقادیر به‌صورت جفت کلید و مقدار استفاده می‌شود.



1. Tuple و List چه تفاوتی دارند؟

**تعریف لیست:**

**لیست** یک نوع داده‌ی قابل تغییر در پایتون است که برای ذخیره‌سازی مجموعه‌ای از آیتم‌ها به کار می‌رود. آیتم‌ها می‌توانند از نوع‌های مختلف باشند و می‌توان آن‌ها را اضافه، حذف یا ویرایش کرد.

**تعریف تاپل:**

**تاپل** یک نوع داده‌ی غیرقابل تغییر در پایتون است که مانند لیست برای نگهداری مجموعه‌ای از داده‌ها به کار می‌رود، اما پس از ایجاد، نمی‌توان مقدار آیتم‌ها را تغییر داد یا آیتم جدیدی اضافه یا حذف کرد.

1. Set در Python چرا برای حذف داده‌های تکراری استفاده می‌شود؟

در Python، نوع داده‌ای set برای حذف داده‌های تکراری استفاده می‌شود چون یک مجموعه (set) تنها مقادیر منحصر به فرد را ذخیره می‌کند. وقتی داده‌ها را به یک set اضافه می‌کنید، به طور خودکار مقادیر تکراری حذف می‌شوند و فقط یک نسخه از هر عنصر باقی می‌ماند. بنابراین، استفاده از set روشی ساده و کارآمد برای حذف داده‌های تکراری است.

1. Stack و Queue چه تفاوتی دارند؟

نفاوت اصلی بین Stack و Queue در نحوه دسترسی به داده‌ها است:

Stack (پشته):

روش دسترسی: "آخرین وارد، اولین خارج" (LIFO - Last In, First Out)

داده‌ها به ترتیب معکوس وارد و خارج می‌شوند. یعنی آخرین عنصری که وارد پشته می‌شود، اولین عنصری است که خارج می‌شود.

مثال: پشته‌ای از کتاب‌ها که آخرین کتابی که گذاشته شده، اول از همه برداشته می‌شود.

Queue (صف):

1. Hash Table چیست و چرا کاربرد دارد؟

Hash Table یک ساختار داده‌ای است که برای ذخیره‌سازی داده‌ها از تابع هش برای تبدیل کلیدها به ایندکس‌های خاص در یک آرایه استفاده می‌کند. این ساختار به دلیل زمان دسترسی سریع (معمولاً O(1) ) برای جستجو، افزودن و حذف داده‌ها کاربرد دارد. از آن در پیاده‌سازی دیکشنری‌ها، کشینگ و حذف داده‌های تکراری استفاده می‌شود.

1. Binary Tree و B-Tree چه تفاوتی دارند؟

Binary Tree:

هر گره حداکثر دو فرزند دارد.

ممکن است نامتعادل باشد.

زمان پیچیدگی بدترین حالت O(n).

B-Tree:

1. چرا Graph Data Structure برای شبکه‌های اجتماعی استفاده می‌شود؟

گراف برای شبکه‌های اجتماعی استفاده می‌شود چون:

* رابطه‌ها را مدل می‌کند: در شبکه‌های اجتماعی، افراد (گره‌ها) به هم مرتبط هستند (یال‌ها). این روابط می‌توانند دوستی، فالو کردن، یا دیگر انواع ارتباطات باشند.
* جستجوی مسیر: می‌توان از گراف برای پیدا کردن مسیرهای کوتاه بین افراد یا محتوای مورد نظر استفاده کرد.
* تحلیل ارتباطات: گراف امکان تحلیل و شناسایی گروه‌های اجتماعی، ارتباطات قوی و شبکه‌های تاثیرگذار را فراهم می‌کند.

1. Dynamic Programming چرا در حل مسائل پیچیده کاربرد دارد؟

**Dynamic Programming (DP)** در حل مسائل پیچیده کاربرد دارد چون:

* **حل زیرمسئله‌ها**: مسائل بزرگ به زیرمسئله‌های کوچکتر تقسیم می‌شوند و نتیجه هر زیرمسئله ذخیره می‌شود تا از محاسبات تکراری جلوگیری شود.
* **کاهش پیچیدگی زمانی**: با استفاده از ذخیره‌سازی نتایج قبلی، پیچیدگی زمانی از O(2n)O(2^n)O(2n) به O(n)O(n)O(n) یا O(n2)O(n^2)O(n2) کاهش می‌یابد.
* **حل مسائل بهینه‌سازی**: در مسائل مانند کوتاه‌ترین مسیر، بیشینه سود و مشابه آن‌ها، DP بهترین جواب را با استفاده از ترکیب نتایج به دست آمده از زیرمسئله‌ها می‌دهد.

1. Recursion چیست و چرا در الگوریتم‌های پیشرفته استفاده می‌شود؟

* Recursion یک تکنیک در برنامه‌نویسی است که در آن یک تابع خود را فراخوانی می‌کند تا مسئله را به زیرمسئله‌های کوچکتر تقسیم کند.
* چرا در الگوریتم‌های پیشرفته استفاده می‌شود؟
* ساده‌سازی حل مسئله: بسیاری از مسائل پیچیده می‌توانند به راحتی با تقسیم به زیرمسئله‌ها و استفاده از فراخوانی‌های بازگشتی حل شوند.
* **نام استاد :مهندس احمدزاده**
* **نام درس :مباحث ویژه**
* **نام ونام خانوادگی :امیرحسین کریم خوی**