

1) ساختمان داده ها چیست و چه کاربردهایی دارد؟

روش هایی برای سازماندهی و ذخیره سازی داده ها هستند که دسترسی و مدیریت آنها را کارآمدتر میکند.

برخی از کاربردهای مهم ساختمان داده ها عبارتند از:

- 1~ جستجو و جستجوی مرتب: ساختمان داده ها به صورت مرتب شده می توانند به سرعت عملیات جستجو را انجام داده و زمان لازم برای یافتن یک داده را به شدت کاهش دهند.
- 2~ پشتیبانی از عملیات درج و حذف: برخی ساختمان داده ها، مانند لیست ها و درخت ها، به صورت موثر عملیات درج و حذف را انجام می دهند و در برخی موارد زمان محاسباتی آن ها به شدت کاهش می یابد.
- 3~ پشتیبانی از عملیات ترتیبی: برخی ساختمان داده ها، مانند صف ها و صفوفونه ها، عملیات ترتیبی مانند اضافه کردن و حذف به صورت موثر را پشتیبانی می کنند.
- 4~ پشتیبانی از عملیات تجمیعی و گروه بندی: ساختمان داده ها می توانند عملیات تجمیعی و گروه بندی را بر روی داده ها انجام دهند و به تجزیه و تحلیل داده ها کمک کنند.
- 5~ پشتیبانی از عملیات گرافی: ساختمان داده های گرافی می توانند به طور موثر عملیات مرتبط با رابطه و ارتباطات بین عناصر را مدیریت کنند و در کاربردهایی مانند شبکه ها، مسیریابی و تجزیه و تحلیل شبکه ها مورد استفاده قرار گیرند.

2) کاربرد، مزایا و معایب آرایه را بیان نمایید.

کاربردهای آرایه:

1. ذخیره سازی مجموعه ای از عناصر هم نوع: آرایه ها برای ذخیره و مدیریت داده های مشابه مانند اعداد یا رشته ها استفاده می شوند.
 2. دسترسی سریع به عناصر: برای جستجو یا دسترسی به داده ها با استفاده از ایندکس (Index) مناسب هستند.
 3. پیاده سازی ساختارهای داده دیگر: آرایه ها معمولاً به عنوان زیرساختی برای پیاده سازی دیگر ساختمان های داده مانند پشته ها و صف ها مورد استفاده قرار می گیرند.
 4. پردازش داده ها: در الگوریتم های مرتب سازی و جستجو، آرایه ها به عنوان ورودی یا خروجی معمولاً بکار می روند.
- مزایای آرایه:

1. دسترسی سریع: زمان دسترسی به عناصر آرایه O(1) است، یعنی می توان به هر عنصر با استفاده از ایندکس به سرعت دسترسی پیدا کرد.
 2. ساده و منظم: ساده ترین نوع ساختمان داده است و برنامه نویسان به راحتی با آن کار می کنند.
 3. فضای ثابت: اندازه آرایه قبل از استفاده تعیین می شود، که قابلیت پیش بینی فضای مورد نیاز را فراهم می آورد.
- معایب آرایه:
1. اندازه ثابت: پس از تعریف اندازه آرایه، نمی توان آن را تغییر داد. اگر نیاز به فضای بیشتری باشد، باید آرایه جدیدی ایجاد کرد و داده ها را کپی کرد.
 2. هزینه حافظه: اگر آرایه کم حجم باشد یا برخی از عناصر آن خالی بمانند، ممکن است فضای هدر رود.
 3. عملیات هزینه بر: عملیات هایی مانند درج و حذف عناصر (به جز در انتها) نیاز به جابجایی عناصر دارند و می توانند زمان بر باشند O(n).

4) کاربرد، مزایا و معایب صف را بیان نمایید.

کاربردهای صف:

1. مدیریت وظایف: در سیستم های عامل، صف ها برای مدیریت دسترسی به منابع و وظایف در صف بندی فرایندها به کار می روند.
 2. حمل و نقل داده: در شبکه های کامپیوتری، صف ها برای مدیریت بسته های داده در طول انتقال به کار می روند.
 3. سیستم های مشتری محور: در برنامه های کاربردی مانند بانک ها و مراکز پشتیبانی، از صف ها برای ترتیب دادن به مشتریان استفاده می شود.
 4. الگوریتم های جستجو: از صف ها در پیاده سازی الگوریتم های جستجوی گسترده (BFS) در گراف ها بهره می برند.
- مزایای صف:
1. سازماندهی منظم: صف ها به ترتیب FIFO (اولین ورودی، اولین خروجی) به مدیریت و سازماندهی وظایف کمک می کنند.
 2. مدیریت ساده: پیاده سازی و استفاده از صف ها به دلیل سادگی و وضوح آنها آسان است.
 3. کاهش پیچیدگی: با استفاده از صف ها، پیچیدگی هایی که ممکن است در هنگام استفاده مستقیم از لیست ها یا آرایه ها بوجود آید کاهش می یابد.
- معایب صف:
1. محدودیت دسترسی: تنها امکان دسترسی به عنصر اول (جلو) و حذف آن وجود دارد، به همین دلیل دسترسی به سایر عناصر دشوار است.
 2. فضای ثابت: اگر صف به صورت آرایه پیاده سازی شود، اندازه آن باید از پیش تعیین شود و ممکن است در صورت پر شدن نیاز به ایجاد فضای جدید باشد.
 3. عملیات زمان بر: عملیات های مانند اضافه کردن یا حذف فضا ممکن است زمان بر باشند، به خصوص در صورت پیاده سازی با لیست های پیوندی.

6) کاربرد، مزایا و معایب پشته را بیان نمایید.

کاربردهای پشته:

1. مدیریت فراخوانی توابع: پشته برای نگهداری داده‌های محلی و اطلاعات مربوط به توابعی که در حال اجرا هستند، استفاده می‌شود (به عنوان مثال، در زبان‌های برنامه‌نویسی).
2. معکوس کردن داده‌ها: می‌توان با استفاده از پشته، به راحتی داده‌ها را معکوس کرد، مانند معکوس کردن یک رشته.
3. عبور از عبارات ریاضی: در محاسبات ریاضی، از پشته برای ارزیابی عبارات و ترتیب عمل‌ها استفاده می‌شود (الگوریتم‌های postfix و prefix).
4. پیاده‌سازی Undo در نرم‌افزارها: در برنامه‌های ویرایش متن و گرافیکی، پشته برای نگهداری عملیات انجام شده و قابلیت بازگشت (Undo) به حالت قبلی بکار می‌رود.

مزایای پشته:

1. سازماندهی ساده: پشته‌ها ساختاری ساده و قابل فهم دارند که می‌توان به راحتی آن‌ها را پیاده‌سازی و مدیریت کرد.
2. عملیات سریع: عملیات اصلی نظیر اضافه کردن و حذف عنصر (Push و Pop) در پشته O(1) زمان می‌برد.
3. مدیریت خودکار فضای حافظه: در بسیاری از زبان‌های برنامه‌نویسی، پشته به‌طور خودکار فضای حافظه را مدیریت می‌کند و از نشتی حافظه جلوگیری می‌کند.

معایب پشته:

1. محدودیت در دسترسی: فقط می‌توان به عنصر بالای پشته دسترسی داشت و دسترسی به عناصر دیگر غیرممکن است.
2. اندازه محدود: ظرفیت پشته معمولاً محدود است، و می‌تواند با Overflow (پر شدن) مواجه شود، مگر اینکه به‌طور پویا مدیریت شود.
3. عدم انعطاف‌پذیری: * به دلیل ساختار LIFO، نمی‌توان عناصر را به صورت تصادفی از پشته حذف کرد، که ممکن است در برخی برنامه‌ها مشکل‌ساز باشد.

8) حداقل تعداد ضرب‌های الزم برای ضرب 4 ماتریس 10×25 ، 10×30 ، 40×10 ، 40×20 را بدست آورید.

اگر دو ماتریس با ابعاد $a \times b$ و $q \times r$ داشته باشیم، تعداد ضرب‌های لازم برای ضرب این دو ماتریس برابر است با $a \times b \times r$.

ضرب اول: $(a \cdot b) \cdot (c \cdot d)$

ضرب $A \cdot b$:

ابعاد: 10×25 و 40×10

تعداد ضرب‌ها: $10000 = 40 \times 10 \times 25$

2 ضرب $C \cdot D$:

ابعاد: 1×40 و 30×1

تعداد ضرب‌ها: $1200 = 30 \times 1 \times 40$

ابعاد حاصل از $A \cdot B$: 25×40

حالا باید این را با $C \cdot D$ که 30×40 است ضرب کنیم.

ضرب نهایی:

ابعاد: $(30 \times 40) \cdot (40 \times 25)$

تعداد ضرب‌ها: $30000 = 30 \times 40 \times 25$

کل تعداد ضرب‌ها

بنابراین، مجموع تعداد ضرب‌ها برای این ترتیب: $41200 = 30000 + 1200 + 10000$