**الگوریتم چیست؟**

مجموعه ای از عملیات مرتب و محدود است که باید به منظور حل مسئله خاصی دنبال می شود. این یعنی زنجیره از دستورالعمل های دقیق وجود دارند که باید به ترتیب خاصی دنبال شوند.

**Stack یا پشته چیست؟**

ساختمان داده ای است که از لیست برای سازماندهی داده ها استفاده می کند.درعین حال از اتنتزاع پشتیبانی می کند.در پشته عمل اضافه و حذف عنصر فقط از یک طرف آن (بالای پشته) انجام می شود.یعنی عنصر که دیر تر ارد شده باشد زودتر از همه خارج می شود.به همین دلیل گفته می شود که پشته از سیاست خروج به ترتیب عکس ورود (Last In-Fast Out) یا به اختصار LIFO پیروی می کند.

عملیات پشته در ساختمان داده ها :

Push: عنصری را به بالای پشته اضافه می کند.

Pop: عنصری را از بالای پشته حذف می کند.

Peek: عنصر بالای پشته را برمیگرداند ولی حذف نمی کند (معادل یک عمل Pop و یک عمل Push).

StackEmpty: خالی بودن پشته را بررسی می کند.

Clear: تمام عناصر پشته را حذف می کند.

Contains: مشخص می کند عنصری در پشته وجود دارد یا خیر.

CopyTo: محتویات پشته را در آرایه ای از نوع Object کپی می کند. در ورودی این متد Index نشان می دهد که آرایه ازچه ایندکسی شروع به کپی شدن شود.

**Queue یا صف چیست؟**

صف یک ساختار داده است که تا حدودی شبیه پشته است. اما بر خلاف پشته، صف از هر دو سمت باز است. از یک سمت همواره برای درج داده و دیگری برای حذف. صف از روش First In-First Out یا به اختصار FIFO استفاده می کند. هر داده ای که اول وارد شد، اول از همه خارج می شود و هر داده ای که آخر از همه وارد شده، آخر از همه خارج می شود.

عملیات صف در ساختمان داده ها:

Enqueue: یک آیتم را به صف اضافه می کند.

Dequeue: یک آیتم را از صف حذف می کند.

Peek: یک عنصر را بدون حذف کردن از ابتدای صف دریافت می کند.

Isfull: پر بودن صف را بررسی می کند.

Isempty: خالی بودن صف را بررسی می کند.

در صف همواره با استفاده از اشاره گر front داده ها را حذف می کنیم و با اشاره گر rear آن ها را اضافه می کنیم.

عملیات Enqueue:

1. بررسی کن که آیا صف پر است یا نه.
2. اگر صف پر بود، خطای Overflow را صادر کرده و خارج شو.
3. اگر صف پر نبود، مقدار اشاره گر rear را یک واحد افزایش بده تا به فضای خالی بعدی اشاره کند.
4. عنصر داده ای را به موقعیت صف که اشاره گر rear نشان می دهد اضافه کن.
5. پیام موفقیت آمیز را برگردان.

عملیات Dequeue:

1. بررسی کن که صف خالی است یا نه.
2. اگر صف خالی بود خطای Overflow صادر کرده و خارج شو.
3. اگر صف خالی نبود، به داده ای که اشاره گر front نشان می دهد دسترسی ایجاد کن.
4. اشاره گر front را یک واحد افزایش بده تا به موقعیت بعدی اشاره کند.
5. پیام موفقیت برگردان.

**Heap چیست؟**

حالت خاصی از ساختمان داده (درخت با نیروی متعادل) است که در آن کلید ریشه-گره (اولین node) با فرزندانش مقایسه می شود و بر همین اساس مرتب می شوند.

دو نوع هیپ وجود دارد:

1. Min-heap: مقدار گره کمتر یا مساوی یکی از فرزندانش است.
2. Max-heap: مقدار گره بزرگتر یا مساوی یکی از فرزندانش است.

الگوریتم ساخت Max-heap:

1. ایجاد یک گره جدید در انتهای هیپ
2. مقدار جدید به گره می دهیم.
3. مقایسه گره با والد
4. اگر والد کمتر از فرزندباشد، تعویض می شوند.
5. گام های 3 و 4 تا زمانی که همه مشخصات هیپ برقرار باشد تداوم می یابد.

الگوریتم حذف Max-heap:

1. گره ریشه را حذف کن
2. آخرین عنصر آخرین سطح را به ریشه جابجا کن.
3. مقدار این گره فرزند را با والدهایش مقایسه کن.
4. اگر مقدار والد کمتر از فرزند بود، جای آن ها را عوض کن
5. گام های 3 و 4 را تا زمانی که مشخصات هیپ برقرار است تکرار کن.

الگوریتم های ساخت و حذف Min-heap مشابه Max-heap است با این تفاوت که مقادیر کمینه به جای بیشینه مورد استفاده قرار می گیرند.