

## پشته و هیپ در علوم کامپیوتر

در علوم کامپیوتر، **پشته** و **هیپ** دو ساختار داده مهم هستند که برای ذخیره سازی و مدیریت اطلاعات استفاده می شوند. هر دو ساختار کاربردهای خاص خود را دارند و در زمینه های مختلفی مورد استفاده قرار می گیرند.

### پشته (Stack)

یک پشته یک ساختار داده **اولین ورودی، اولین خروجی (FIFO)** است. به این معنی که اولین عنصری که به پشته اضافه می شود، اولین عنصری است که از آن خارج می شود. پشته ها را می توان با استفاده از لیست های پیوندی یا آرایه ها پیاده سازی کرد.

### کاربردهای پشته:

- **بازگشت تابع:** پشته ها برای ذخیره سازی آدرس بازگشت تابع ها در هنگام فراخوانی توابع تو در تو استفاده می شوند.
- **تجزیه و تحلیل عبارت:** پشته ها در تجزیه کننده های زبان برای ذخیره سازی نمادهای ورودی و قوانین گرامری استفاده می شوند.
- **بیان معکوس لهستانی:** پشته ها برای محاسبه عبارات ریاضی به صورت بیان معکوس لهستانی استفاده می شوند.

### هیپ (Heap)

یک هیپ یک ساختار داده **درخت کامل باینری** است که دارای **خاصیت توده (heap property)** است. خاصیت توده به این معنی است که کلید هر گره باید بزرگتر یا مساوی با کلید هر یک از فرزندانش باشد. هیپ ها را می توان با استفاده از آرایه ها پیاده سازی کرد.

## کاربردهای هیپ:

- الگوریتم های مرتب سازی: هیپ ها در الگوریتم های مرتب سازی مانند مرتب سازی هیپ و مرتب سازی درجا استفاده می شوند.
- انتخاب اول: هیپ ها در الگوریتم های انتخاب اول مانند الگوریتم Prim برای یافتن درخت پوشای کمترین وزن استفاده می شوند.
- ذخیره سازی پویا: هیپ ها برای مدیریت حافظه پویا در زبان های برنامه نویسی مانند ++C استفاده می شوند.

## تفاوت های کلیدی بین پشته و هیپ:

- ساختار: پشته ها به صورت لیست های پیوندی یا آرایه ها پیاده سازی می شوند، در حالی که هیپ ها به صورت درختان کامل باینری پیاده سازی می شوند.
- دسترسی: در پشته ها، فقط می توان به عنصر بالا دسترسی داشت و آن را از پشته خارج کرد. در هیپ ها، می توان به هر عنصری در درخت دسترسی داشت.
- عملیات: عملیات اصلی در پشته ها فشار (push) و پاپ (pop) هستند. عملیات اصلی در هیپ ها درج (insert) و حذف (delete) هستند.
- کاربرد: پشته ها برای بازگشت تابع، تجزیه و تحلیل عبارت و بیان معکوس لهستانی استفاده می شوند. هیپ ها برای مرتب سازی، انتخاب اول و ذخیره سازی پویا استفاده می شوند.

## مثال:

فرض کنید می خواهید لیستی از اعداد را به ترتیب صعودی مرتب کنید. می توانید از مرتب سازی هیپ برای این کار استفاده کنید. ابتدا اعداد را به هیپ اضافه کنید. سپس، به طور مکرر بزرگترین عدد را از هیپ حذف کنید و آن را به لیست مرتب شده اضافه کنید. این فرآیند را تا زمانی که تمام اعداد از هیپ حذف شوند، ادامه دهید.

در اینجا یک مثال از نحوه مرتب سازی لیست [5, 2, 4, 1, 3] با استفاده از مرتب سازی هیپ آورده شده است:  
ایجاد هیپ:

[5][2, 5][4, 2, 5][1, 4, 2, 5][3, 1, 4, 2, 5]

حذف بزرگترین عدد و اضافه کردن آن به لیست مرتب شده:

[3][4, 2][1, 4, 2][5, 1, 4, 2]

حذف بزرگترین عدد و اضافه کردن آن به لیست مرتب شده

[1, 3]: [4, 2][5, 4, 2]