



ساختمان داده‌ها (۲۲۸۲۲)

مدرس: حسین بومری

[زمستان ۹۹]

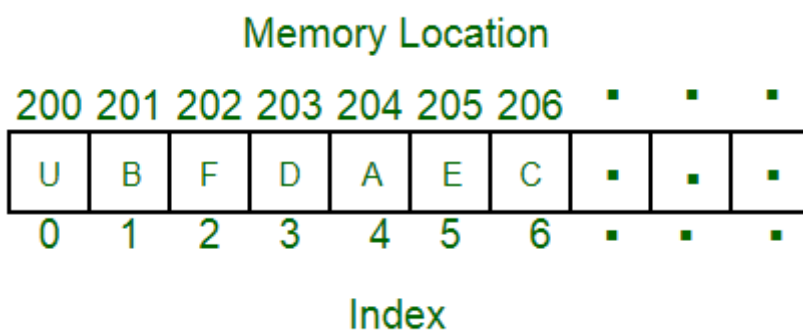
نگارنده: مبین معدنی

جلسه ۹: آشنایی مقدماتی با داده ساختارها

در این جلسه قصد داریم تا با داده ساختارهای مقدماتی آشنا شویم و توابع مختلف را از نظر زمانی با هم مقایسه کنیم

۱ آرایه Array

داده ساختار آرایه از یک پونتر شروع و طول آرایه مانند شکل زیر ساخته شده است :



مستقیم آدرس اندیس را صدا میزنیم	$O(1)$	element i
	$O(n)$	search
	$O(n)$	index delete
جای مورد نظر را پیدا میکنیم بعد بقیه را یک واحد شیفت میدهیم	$O(n)$	index insert
	$O(1)$	front delete
	$O(n)$	front add
	$O(n)$	min
	$O(n)$	max
با روشی به اسم median که در جلسات بعد بررسی خواهد شد	$O(n)$	i امین عنصر مرتب
	$O(n)$	min delete
	$O(n)$	build

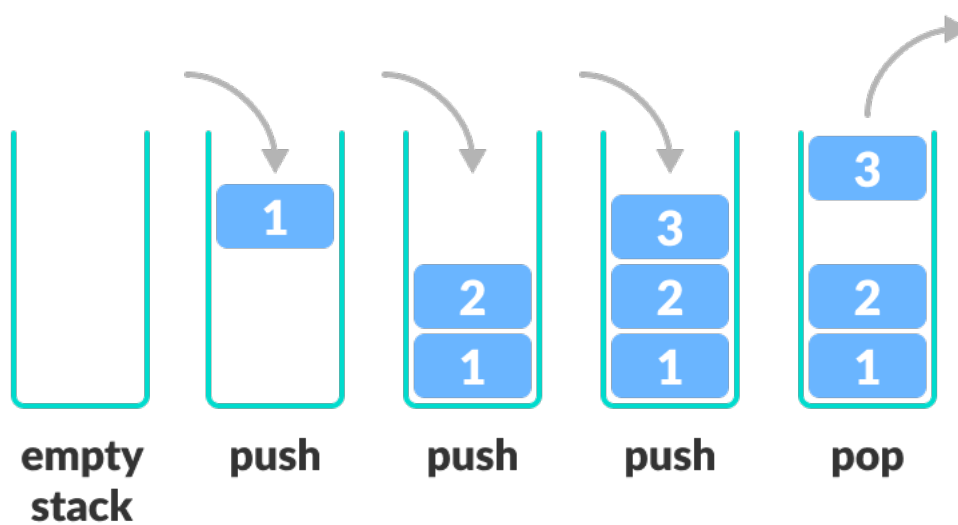
۱.۱ آرایه مرتب sort Array

این داده ساختار نیز مانند داده ساختار قبلی است و تنها تفاوتش در این است که اطلاعات در آن به صورت مرتب نگهداری میشود.

مستقیم آدرس اندیس را صدا میزنیم	$O(1)$	element i
به صورت باینری سرچ	$O(\log n)$	search
	$O(n)$	index delete
جای مورد نظر را پیدا میکنیم بعد بقیه را یک واحد شیفت میدهم	$O(n)$	index insert
	$O(1)$	front delete
بی معنی است	—	front add
	$O(1)$	min
	$O(1)$	max
	$O(1)$	i امین عنصر مرتب
	$O(1)$	min delete
	$n) O(n \log$	build

۲ پشته Stack

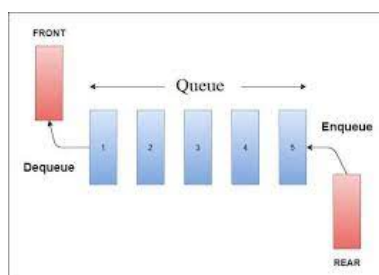
پشته با شعار $LIFO = \text{last in first out}$ نوعی داده ساختار است که آخرین داده ی ورودی در آن اول خارج میشود مانند شکل زیر:



$O(n)$	element i
$O(n)$	search
$O(n)$	index delete
$O(n)$	index insert
حذف از بالا $O(1)$ ، از پایین $O(n)$	front delete
حذف از بالا $O(1)$ ، از پایین $O(n)$	front add
$O(n)$	min
$O(n)$	max
$O(n)$	i امین عنصر مرتب
$O(n)$	min delete
$O(n)$	build

۳ صف queue

صف با شعار $FIOF = \text{first in first out}$ داده ساختاری است که در آن داده ای که اول وارد میشود ، اول نیز خارج میشود ، مانند شکل زیر :



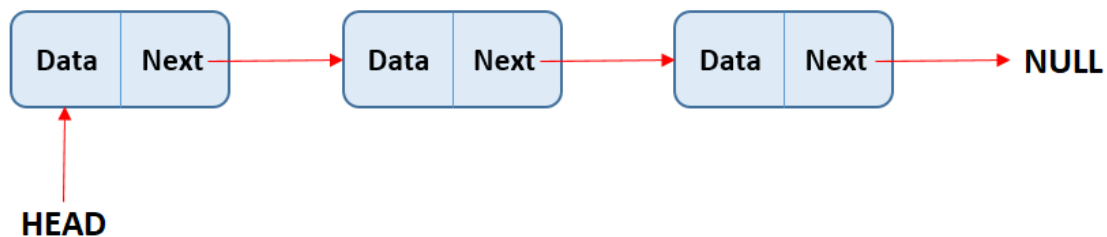
$O(n)$	element i
$O(n)$	search
$O(n)$	index delete
$O(n)$	index insert
اول صف $O(n)$ ، آخر صف $O(1)$	front delete
اول صف $O(n)$ ، آخر صف $O(1)$	front add
$O(n)$	min
$O(n)$	max
$O(n)$	i امین عنصر مرتب
$O(n)$	min delete
$O(n)$	build

۴ لیست List

عناصر را به ترتیب وارد کردن نگهداری میکند ولی لزوماً در حافظه پشت سر یکدیگر نگهداری نمیشوند.

۱.۴ لیست پیوندی Linked List

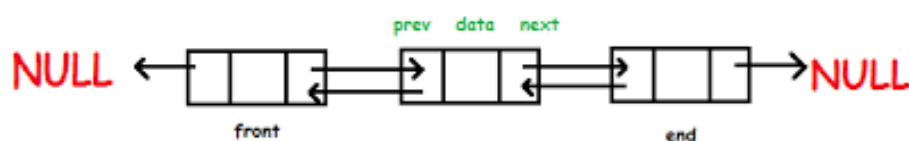
در این داده ساختار اطلاعات به صورت دسته هایی از node نگهداری میشود که هر node مانند شکل زیر دارای دو بخش است



	$O(n)$	element i
	$O(n)$	search
	$O(n)$	index delete
فرض میکنیم که روی node مورد نظر قرار داریم و حال میخواهیم عملگر را اعمال کنیم	$O(1)$	index insert
فرض میکنیم که روی node مورد نظر قرار داریم و حال میخواهیم عملگر را اعمال کنیم	$O(1)$	front delete
فرض میکنیم که روی node مورد نظر قرار داریم و حال میخواهیم عملگر را اعمال کنیم	$O(1)$	front add
	$O(n)$	min
	$O(n)$	max
	$O(n)$	i امین عنصر مرتب
	$O(n)$	min delete
	$O(n)$	build

۲.۴ لیست پیوندی دوسویه doubly linkedlist

تنها تفاوتش با داده ساختار قبلی این است که node ها هم به عنصر قبلی و هم به عنصر بعدی اشاره میکنند.



	$O(n)$	element i
	$O(n)$	search
	$O(1)$	index delete
فرض میکنیم که روی node مورد نظر قرار داریم و حال میخواهیم عملگر را اعمال کنیم	$O(1)$	index insert
فرض میکنیم که روی node مورد نظر قرار داریم و حال میخواهیم عملگر را اعمال کنیم	$O(1)$	front delete
فرض میکنیم که روی node مورد نظر قرار داریم و حال میخواهیم عملگر را اعمال کنیم	$O(1)$	front add
	$O(n)$	min
	$O(n)$	max
	$O(n)$	i امین عنصر مرتب
	$O(n)$	min delete
	$O(n)$	build

جمع بندی

	آرایه	آرایه مرتب	پشته	صف	لیست پیوندی	لیست پیوندی دو س
element i	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
search	$O(n)$	$O(\log n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
index delete	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$
index insert	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$
front delete	$O(1)$	$O(1)$	$O(n)$ ، $O(1)$	اول صف $O(n)$ ، آخر صف $O(1)$	$O(1)$	$O(1)$
front add	$O(n)$	—	$O(n)$ ، $O(1)$	اول صف $O(n)$ ، آخر صف $O(1)$	$O(1)$	$O(1)$
min	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
max	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
i امین عنصر مرتب	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
min delete	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
build	$O(n)$	$O(n \log n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$