

نکته: همراه بھترین میادن ساز برای مجموعہ مستاھن، مناسب با کاربرد و داشت  
یہ عملیات صورت سُیقیلیتی لھستا۔

قرداد: میر عنصر مجموعہ بروایتھن از اصطلاح عنصر طاسی استفادہ ہی لئے۔  
یا در اقسام عملیات از اصطلاح مجموعہ یا مجموعہ بروایا بے جار مجموعہ بروایتھن

برائقتصار،  
استفادہ ہی لئے، در این درس، تمام مجموعہ ها را مستاھن فرغ ہی لئے۔  
یعنی کو تعریف ہی لئے  
بعبارت دیگر،

\* عنصر: هر عنصر طاسی از نکل مجموعہ حاوی رہا یا از حصیقہ عاست کہ بہ طور محض  
کوئی از آنها یہ عنوان کنیے بڑاں میں در تظریق تھے ہی سود لازم ہے ذرا است کہ  
این اعکان بیزد ہو ڈکر کہ چنین حصیقہ کلیوی بارے ہی مجموعہ بروایی در تظریق تھے  
سُود و مثود رئی بیرون کلیوی ہو ڈون حصیقہ ما و محدود نہ ہو۔

\* عملیات روی مجموعہ بروایا: بہ طور کل عملیات روی مجموعہ بروای راجح قابل در دردہ زیر  
بیان گرد:

\* سرکال ہای بازیابی (Retrieval query) (در اطلاعاتی):  
را در صورت مجموعہ برجی گراندہ

\* پرسچال ہای مرغز سانی (Update query): بسبت

ایجاد تغییر (حذف و اضافہ عنصر مجموعہ) در مجموعہ ہی سُود

برخی از متدات ترین این عملیات‌ها در ارائه کورده شده است:

• یک پرسان بزرگی بازیابی است و به صورت زیر عمل می‌کند:

Search( $S, k$ )

\* آرسی کلید  $k$  در مجموعه  $S$  وجود داشته باشد، آن سی را بروز راند.

\* در عین تصورت، Null برعی برداشت.

• یک پرسان بروزرسانی است و عنصر  $x$  را به مجموعه  $S$  اضافه می‌کند.

Insert( $S, x$ )

• یک پرسان بروزرسانی است و عنصر  $x$  را از مجموعه  $S$  حذف می‌کند.

Delete( $S, x$ )

• یک پرسان بازبایی است و سی (یا سی‌های) از مجموعه  $S$  کو را کمترین مقدار را درد، برعی برداشت.

Minimum( $S$ )

• یک پرسان بازبایی است و سی (یا سی‌های) از مجموعه  $S$  کو را کمترین مقدار را درد، برعی برداشت.

Maximum( $S$ )

• یک پرسان بازبایی است که مقدار از سی کو بزرگتر از سی کو (براسنده کلید سی) در مجموعه  $S$  را برعی برداشت. نازم به ذکر اس که آرسی کو حائز یکم باشد، Null را برعی برداشت.

Successor( $S, x$ )

• آرسی کو حائز یکم باشد، Null را برعی برداشت.

= Predecessor(S, x)

یک پرسچان بازیاب است که نزدیکترین سی مرحله از سی خ  
 (برآسان کنیش) در مجموع دل را برای گرداند. لازم به ذکر  
 است که اگر عضو صنعتیم باشد Null را باید گرداند.

با (سیاره از توالی بالا،

اطلاعات تفصیلی: برای مرتب کردن مجموع کن! عضوی از کن  
 (نامه تکلیف) مصلحت ر:

۱. یکبار فراخوانی Minimum(S)

۲. Successor n-1 بار فراخوانی

۳. اولین کوچکترین  $\rightarrow \text{Minimum}(S) \rightarrow x_1$

۴. دومن کوچکترین  $\rightarrow \text{Successor}(S, x_1) \rightarrow x_2$

⋮

۵. i-اعنی کوچکترین  $\rightarrow \text{successor}(S, x_{i-1}) \rightarrow x_i$

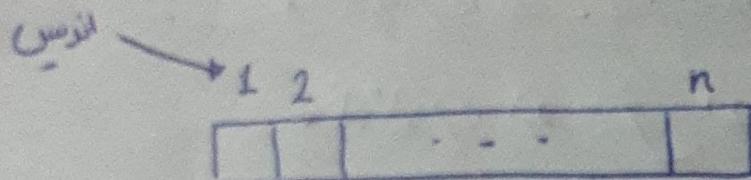
⋮  
۶. n-اعنی کوچکترین  $\rightarrow \text{successor}(S, x_{n-1}) \rightarrow x_n$

در ادامه به معنی بخش از راهه ساختار هار یا به تغیر لیست پیوندی، صفات سپهه و

درخت رسی دار خواهیم پرداخت.

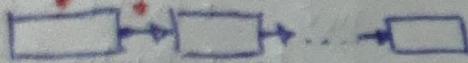
## لیست سیویندی (Linked List)

لیست: یک داره ساختار است که اسیای آن در ترتیب خطی (متصل با ازین لیست) مرار گرفته اند.



نماینده از اینکه لیست

لیست پیویندی: یک داره ساختار است که اسیای آن در ترتیب خطی (متصل با ازین لیست) مرار گرفته اند. این را می‌توان (اسایه و عطفه) املاه کرد. معرفه = نزدیک (اسایه و عطفه) املاه کرد. معرفه = نزدیک (اسایه و عطفه) املاه کرد.



نماینده از اینکه لیست پیویندی

لغه: متغیر را ترتیب خطی، ترتیب در جایگاه هر تیری عناصر و ترتیب بر اساس مکانی عناصر است.

انواع لیست پیویندی:

\* لیست پیویندی که مفهوم (singly linked list)

\* لیست پیویندی دو مفهوم (Doubly linked list)

\* لیست پیویندی دایره ای (circular linked list)

در ادامه، اینجا لیست پیویندی دو مفهوم را معرفی می‌کنیم و سپس با آنرا می‌دانیم. آن، لیستهای پیویندی را معرفی می‌کنیم.

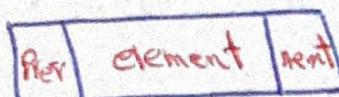
لیست پیویزی در طرفه:

هر نود در این لیست شامل سه بخش زیر است:

\* اسکاراً گریه نزد قبلی که آن را  $\text{prev}$  نشان می‌دهیم،

\* اسکاراً گریه نزد بعدی که آن را  $\text{next}$  نشان می‌دهیم،

\* علاوه بر این دو داده از عضویت از مجموعه آن را  $\text{element}$  نشان می‌دهیم.



خاصیت نود در لیست پیویزی نویفرم

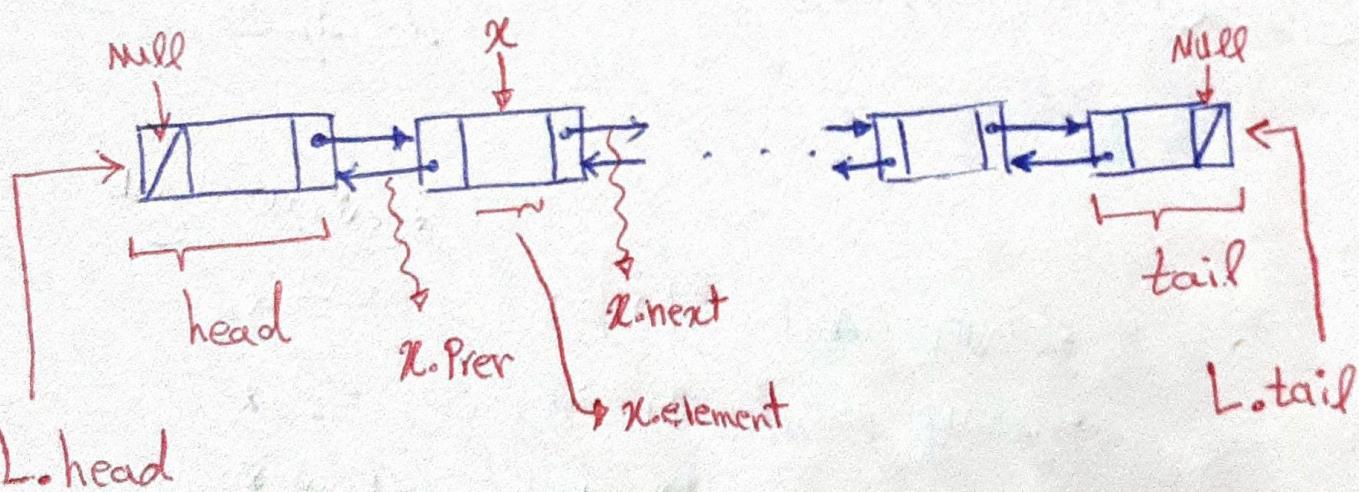
Record node {

element : datatype,

prev : Pointer to node,

next : Pointer to node }

سلک زیر مذکور به از لیست را که شامل چندین نور است، نشان می‌دهد:



Record List {

size : integer,

head : Pointer to node,

tail : Pointer to node }

پیش نکات در رایله با لیست پیوندی دو طرفه:

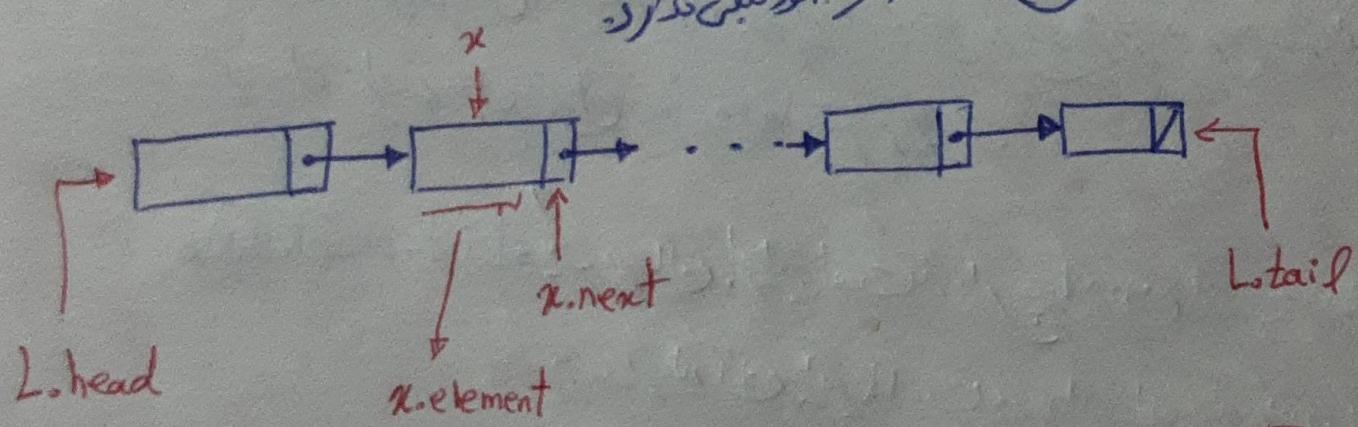
۱ = اگر  $L.\text{head} = \text{NULL}$  باشد، برعنهای متناسب که لیست نباشد است.

۲ = اگر  $x.\text{next} = \text{NULL}$  باشد، عضور  $x$  در ترتیب خطا لیست، عضور آخر است.

۳ = اگر  $x.\text{prev} = \text{NULL}$  باشد، عضور  $x$  در ترتیب خطا سهیل (نهنده) لیست، عضور اول است.

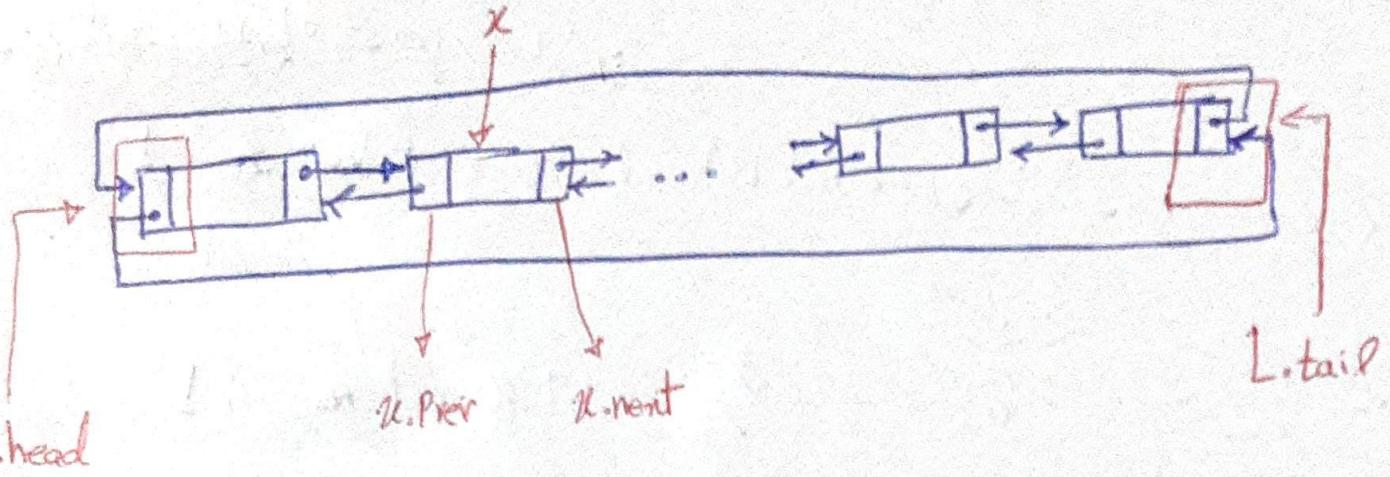
لیست پیوندی دو طرفه:

سبسیم به لیست پیوندی دو طرفه است، با این تفاوت که هر یوزر کان نهایا شامل اسما و گزینه نود بعدی است را میگیرد اگر به نود قبلی دارد.



لیست پیوندی دایره‌ای:

این نوع لیست قابل تعریف برای لیست‌های سیزدهم یکدیگر دو طرفه را دو طرفه است. با اینحال در ازایه تنها یک لیست پیوندی دایره‌ای برای حالت دو طرفه می‌برازیم.



سؤال: مرا درین ساختار نیاز فست `tail` درست گهاده شود. هنر، از طرف `head` سورت

$L.tail = L.head.prev$ : او بروی تکان برآن دسترسی است

عملیات روی لیست های پیوندی:

کلیه عملیاتی در درین بخش عرفی می شوند مگراین لیست پیوندی دو طرفه شرح شده است،  
با این حال برای آن کابل تبدیل برگشته مانند می باشد.

\*: یک درمان پاژایی است که برای کدن اولین عضوی

که مقدار ملید آن برابر ۰ است مورد انتقاده قرار گیرد. اگر

عنصر در لیست موجود یا نداشته باشد عنصر بزرگتر از همی شود، در

غیر از قصورت مقدار `NULL` خروجی خواهد بود.

## List-search(L, x)

1.  $x = L.\text{head}$
2. while ( $x \neq \text{Null}$  and  $x.\text{key} \neq k$ ) do
3.      $x = x.\text{next}$
4. return  $x$

پیچیدگر زمانی الگوریتم مفروض در بیشترین حالت  $O(n)$  است، جایی که  $n$  تعداد اندیشورهاست.  
لیست پیوندی است.

: یک درس همان بجز رسانی است که عصر x را به انتخابی لیست L  
افزایش می‌کند.

## List-Insert(L, x)

1.  $x.\text{next} = L.\text{head}$
2. if ( $L.\text{head} \neq \text{Null}$ ) then
3.      $L.\text{head} = \text{Prev} = x$
4.  $L.\text{head} = x$
5.  $x.\text{Prev} = \text{Null}$

پیچیدگر زمانی الگوریتم مفروض  $O(1)$  است.