فایل ششم درس ساختمان های داده

- فایل آماده شده برای جلسه ادامه مبحث ساختار لیست پیوندی و کاربردهای آن و شروع مبحث درخت و صرفا برای کلاس آنلاین در این ترم می باشد.

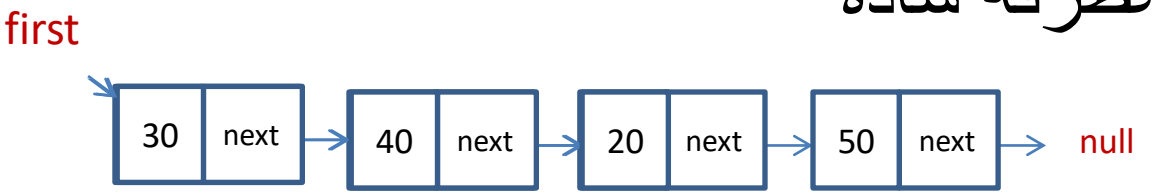
همواره سلامت و موفق باشید- اسبقی

انواع لیست پیوندی

- لیست پیوندی یکطرفه ساده
- لیست پیوندی یکطرفه حلقوی
- لیست پیوندی با گره رأس
- لیست پیوندی دوطرفه ساده
- لیست پیوندی دوطرفه حلقوی

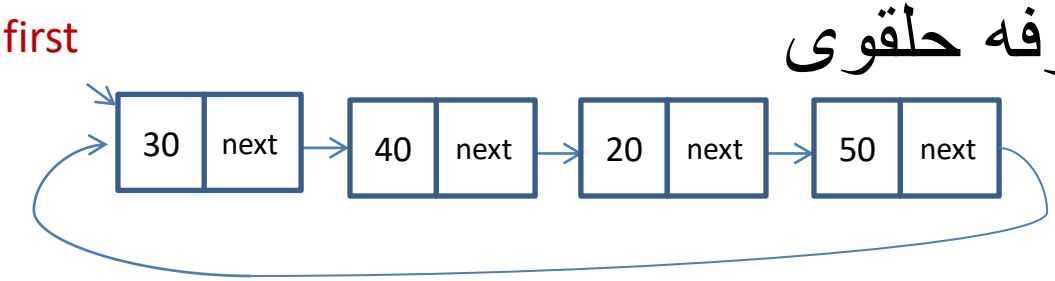
لیست پیوندی یک طرفه

- لیست پیوندی یکطرفه ساده



if(temp->next ==null)

- لیست پیوندی یکطرفه حلقوی

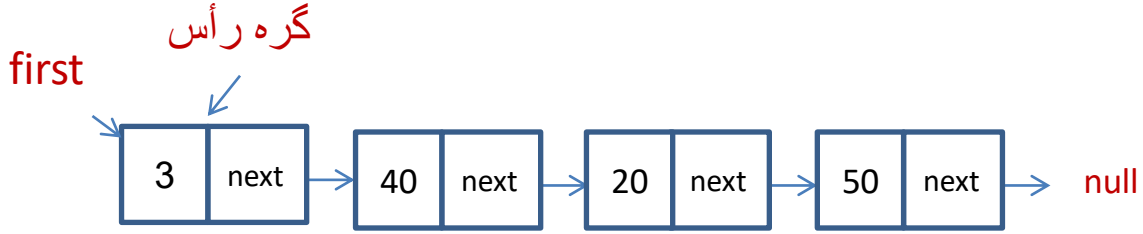


if(temp->next ==first)

لیست پیوندی با گره رأس

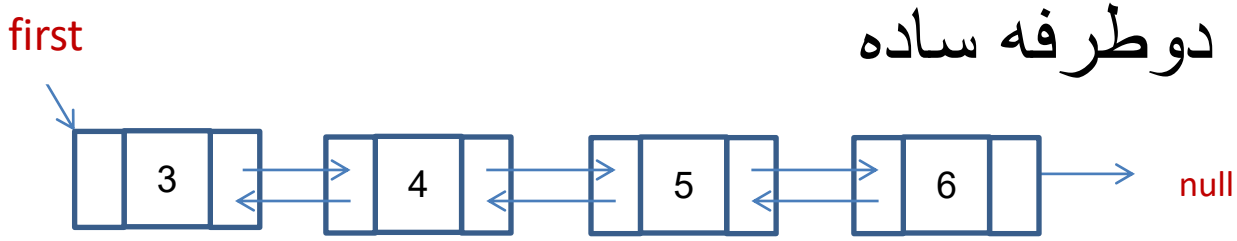
- در لیست پیوندی با گره رأس، یگ گره برای نگهداری تعداد گره های لیست در نظر گرفته می شود.
- مزیت: همواره تعداد گره های لیست به راحتی و بدون پیمایش آن، در دسترس است.
- عیب:

هر عمل درج یا حذف از لیست، نیازمند بروزرسانی مقدار گره رأس می باشد.

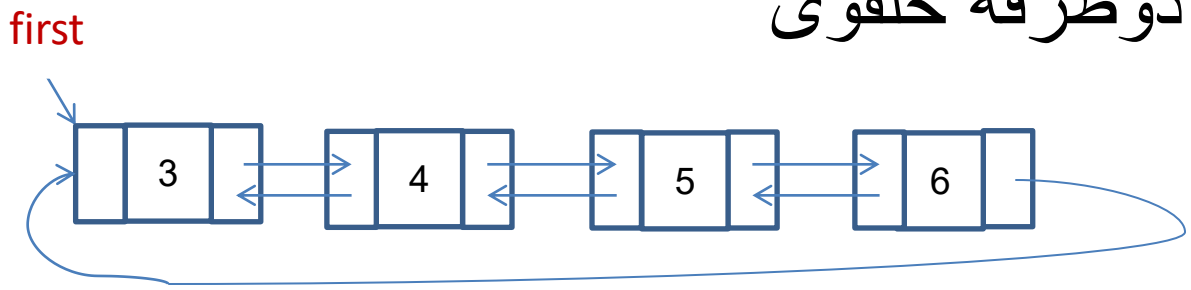


لیست پیوندی دو طرفه(لیست دو پیوندی)

- لیست پیوندی دوطرفه ساده



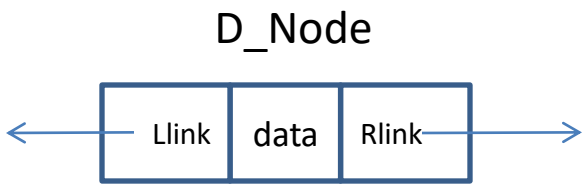
- لیست پیوندی دوطرفه حلقوی



گره لیست دو پیوندی

یک گره لیست دو طرفه دارای دو اشاره گر به گره های بعدی و قبلی می باشد.

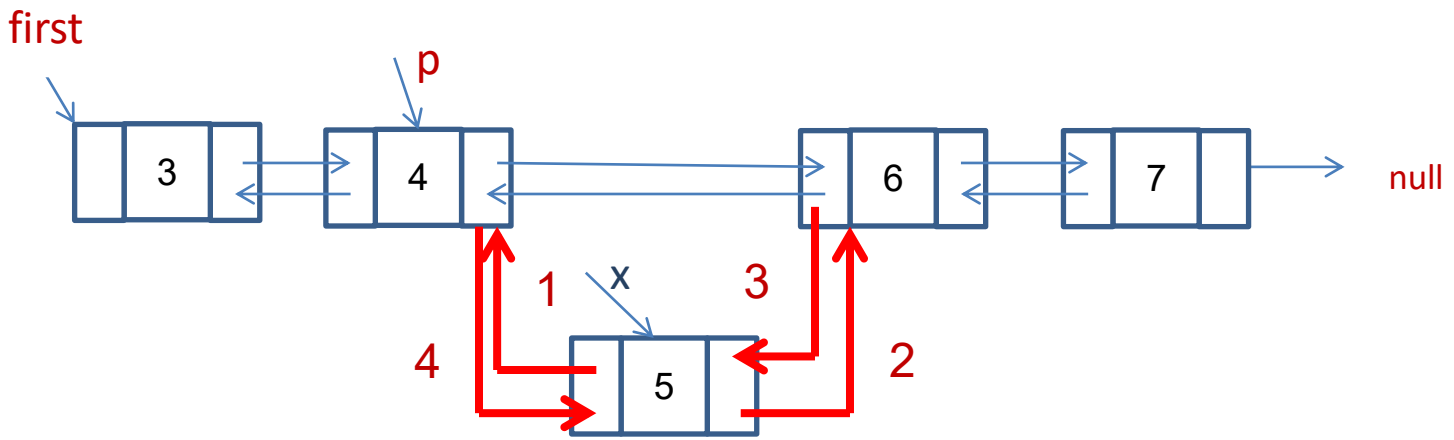
```
struct  D_Node{
int   data;
D_Node   *Rlink;
D_Node   *Llink;};
```



کاربرد:

استفاده در ساختارهایی که نیاز است به عنصر بعدی و قبلی دسترسی داشته باشیم.

درج عنصر در لیست دو طرفه



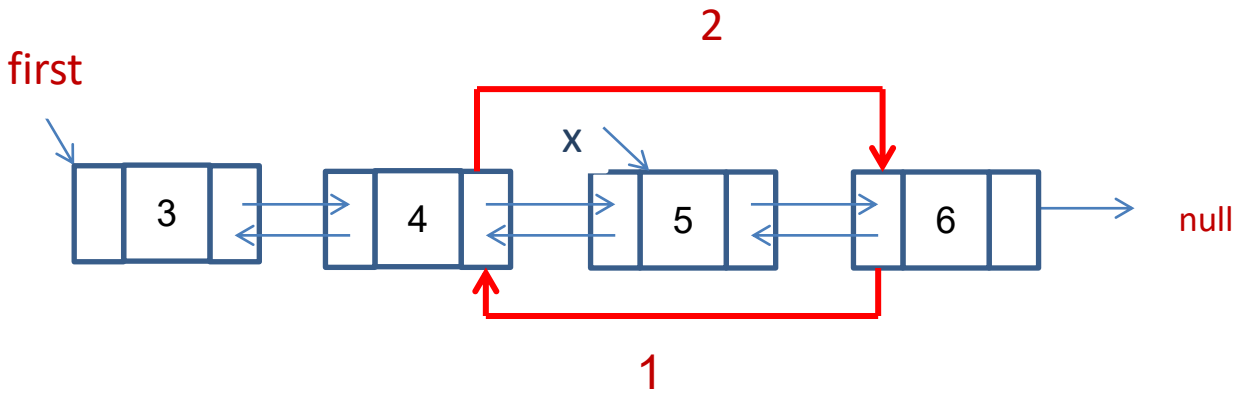
1)x->Llink = p;

2)x->Rlink = p->Rlink;

3)p->Rlink ->Llink = x;

4)p->Rlink = x;

حذف عنصر از لیست دو طرفه

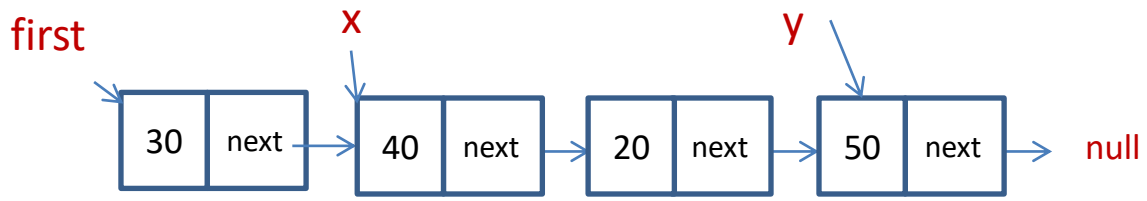


1) x->Rlink->Llink = x->Llink;

2) x->Llink->Rlink = x->Rlink;

مثال

- در لیست پیوندی زیر دو گره x و y را جابجا نمایید.



1)y->next = x->next;

2)x->next ->next =x;

3)x->next = null;

4)first->next = y;

برخی کاربردهای لیست پیوندی

- پیاده سازی پشته با استفاده از لیست پیوندی
- پیاده سازی صف با استفاده از لیست پیوندی
- استفاده از لیست پیوندی در جمع اعداد صحیح بزرگ
- استفاده از لیست پیوندی در نمایش چند جمله ایهای اسپارس

استفاده از لیست پیوندی در جمع اعداد صحیح بزرگ

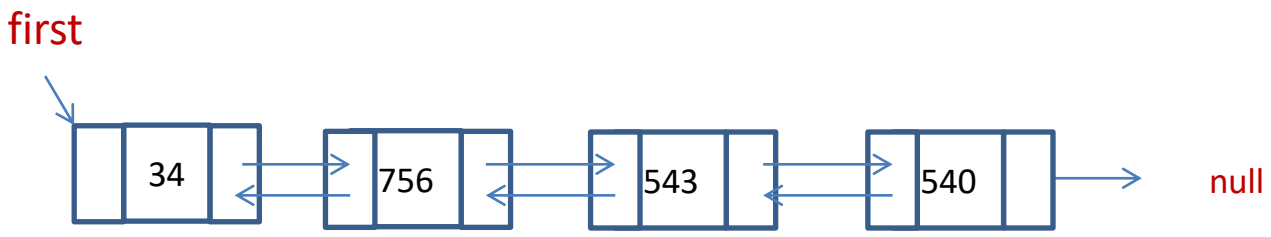
برای ذخیره اعداد صحیح در فضای حداکثر 4 بایتی داریم:

$4byte = 32\ bite$ 2^{32} - 1 = 2,174,483,647

بنابراین اگر نیاز به ذخیره عددی بزرگتر از بزرگترین عدد بالا وجود داشته باشد، بایستی از لیست های پیوندی بصورت زیر استفاده گردد:

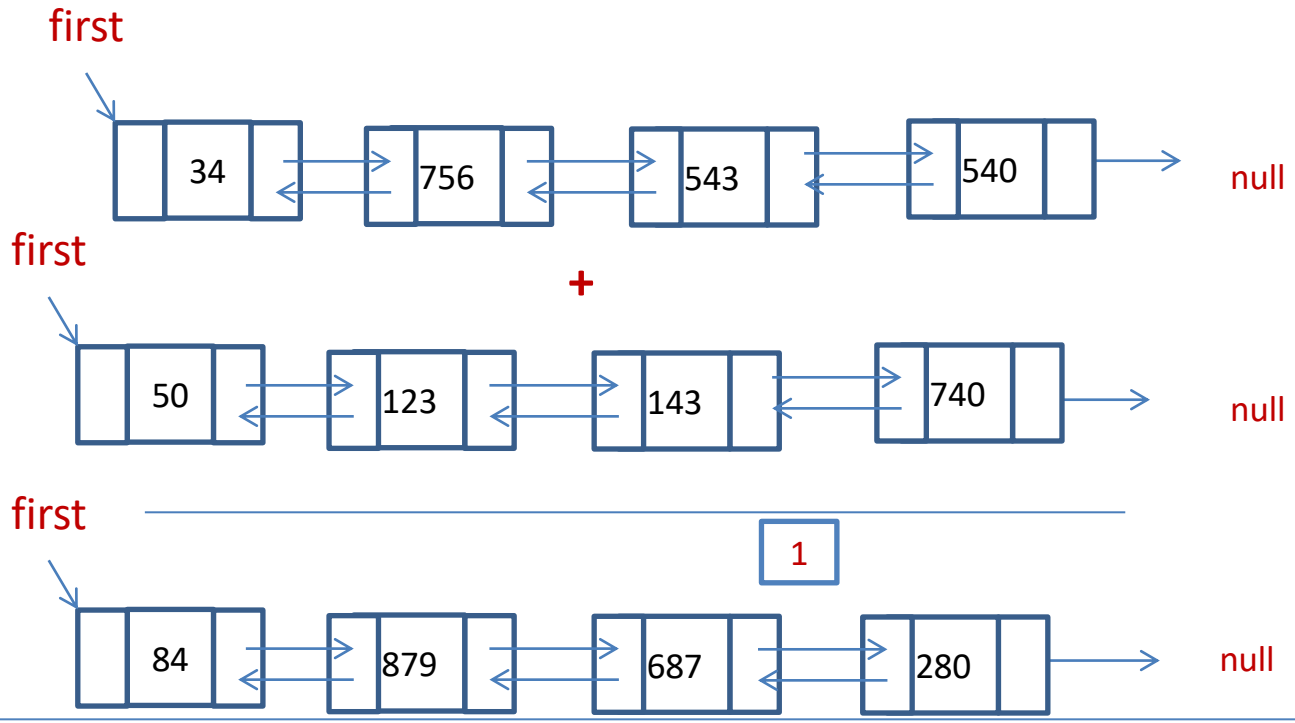
X = 34,765,543,540

از چه نوع لیست پیوندی بایستی استفاده گردد؟ چرا؟



استفاده از لیست پیوندی در جمع اعداد صحیح بزرگ

حال اگر لازم باشد بعنوان مثال عمل جمع دو عدد صحیح بزرگ بصورت زیر انجام پذیرد، از آنجاییکه نیاز به استفاده از carry وجود دارد، پس باید از لیست دوپیوندی استفاده شود.



چند جمله ای

- چند جمله ای

$$f(x) = a\,x^{0} + a_1 x^{1} + a_2\,x^{2} + a_3\,x^{3} + \ldots\ldots\ldots + a_n\,x^{n}$$

روش های پیاده سازی:

-استفاده از آرایه

-استفاده از لیست پیوندی

پیاده سازی چندجمله ای با استفاده از آرایه

$f(x) = 5 + 7x^2 + 3x^4$

0	1	2	3	4
5	0	7	0	3

$f(x) = 9 + 7x^3 + 36x^7$

0	1	2	3	4	5	6	7
9	0	0	7	0	0	0	36

$f(x) = 2 + 3x^{99}$

0	1	2	.....	99
2	0	0	0	0	3

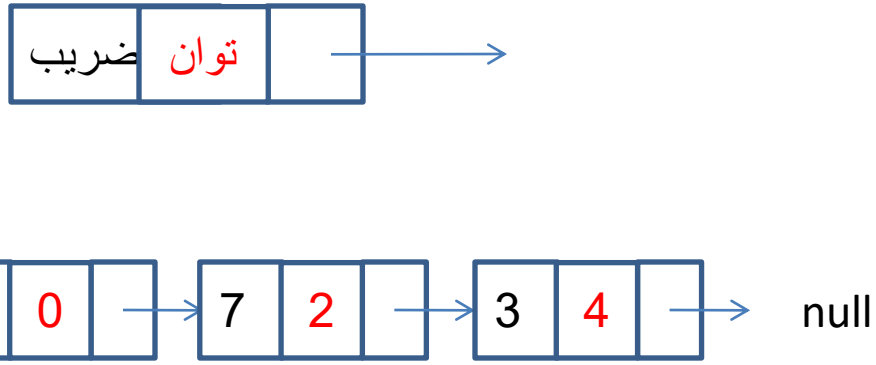
مشکل پیاده سازی چند جمله ای با آرایه؟

پیاده سازی چندجمله ای با استفاده از لیست پیوندی

$f(x) = 5 + 7x^2 + 3x^4$

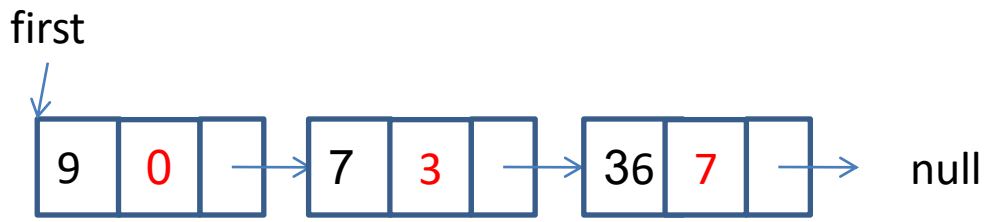
در پیاده سازی با لیست پیوندی، ابتدا باید نوع لیست پیوندی مورد نیاز را مشخص نماییم:

لیست پیوندی با دو محل برای ذخیره توان و ضریب و یک اشاره گر برای اشاره به گره بعدی

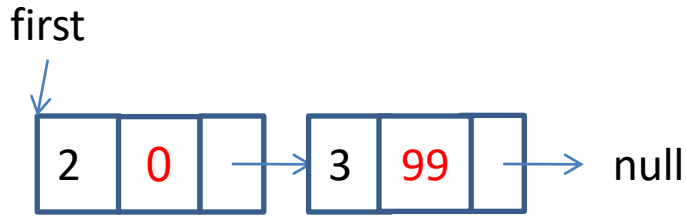


پیاده سازی چندجمله ای با استفاده از لیست پیوندی

$f(x) = 9 + 7x^3 + 36x^7$



$f(x) = 2 + 3x^{99}$



پیاده سازی چندجمله ای با استفاده از لیست پیوندی

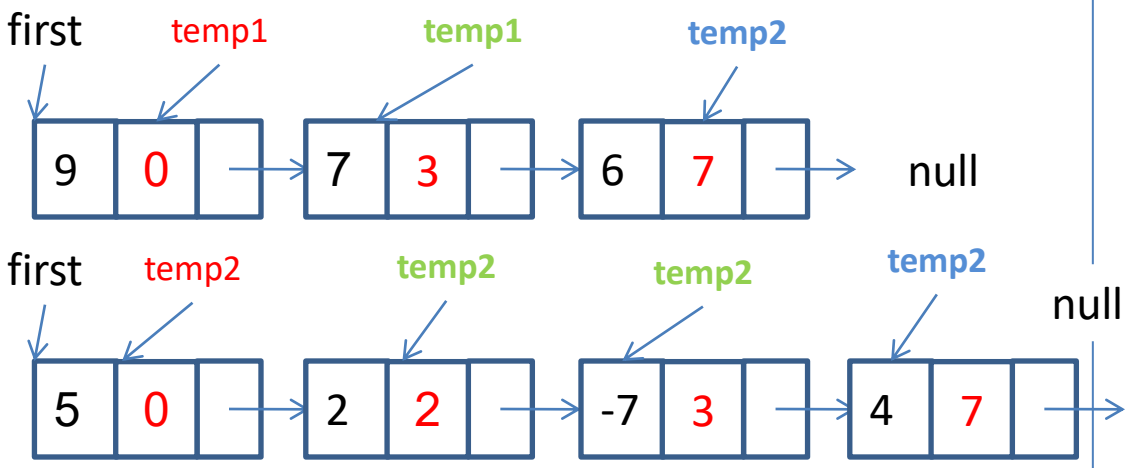
$f(x) = 9 + 7x^3 + 6x^7$

+

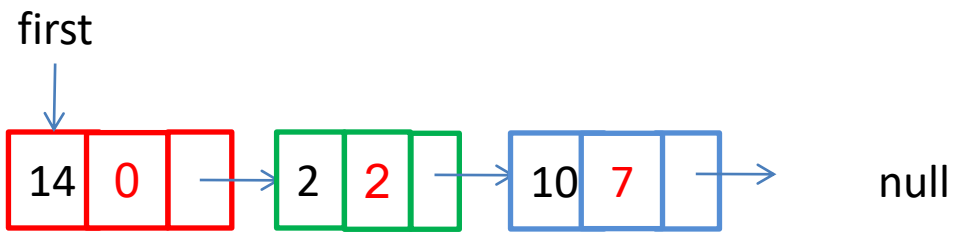
$f(x) = 5 + 2x^2 - 7x^3 + 4x^7$

$f(x) = 14 + 2x^2 + 0x^3 + 10x^7$

$f(x) = 14 + 2x^2 + 10x^7$

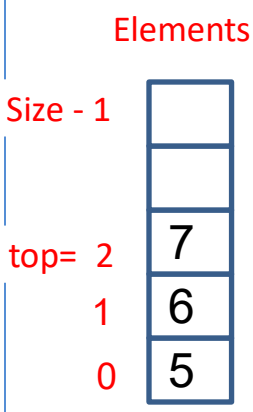


برای هر لیست یک اشاره گر در نظر گرفته می شود که در سطح گره های لیست اشاره میکنند:
اگر توان دو گره از دو لیست مساوی باشد: ضرایب آنها جمع شده و درصورتیکه صفر نباشد بعنوان گره جدید در لیست نتیجه اضافه می شود.
اگر توان ها نامساوی باشد: گره با توان کوچکتر در نتیجه اضافه شده و گره بعدی لیست مجددا مقایسه می گردد.
درصورت اتمام گره های یک لیست، لیست دوم در نتیجه اضافه می گردد.

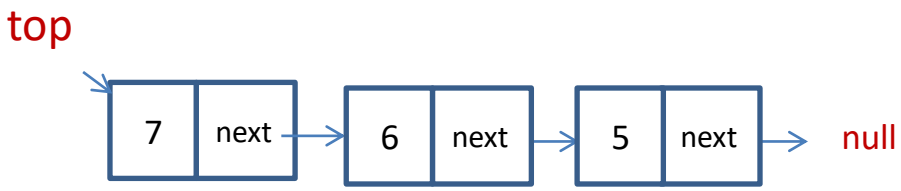


پیاده سازی پشته با استفاده از لیست پیوندی

- در پیاده سازی پشته با آرایه داشتیم:

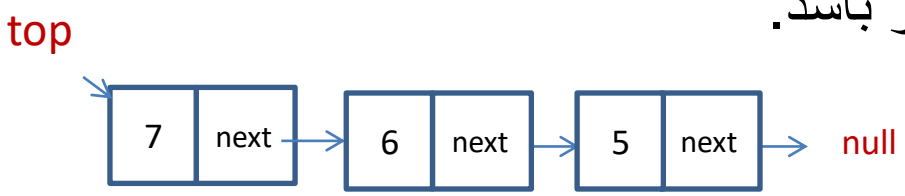


- حال اگر عناصر این پشته را بصورت غیر خطی با استفاده از لیست پیوندی پیاده سازی نماییم، داریم:



پیاده سازی پشته با استفاده از لیست پیوندی

در پیاده سازی پشته با لیست پیوندی، عملیات درج(push) و حذف(pop) از پشته، بایستی هر دو از ابتدای لیست پیوندی انجام پذیرد تا خاصیت و رفتار پشته مبنی خروج به ترتیب عکس ورود برقرار باشد.



درج عنصر در پشته(push): درج در ابتدای لیست

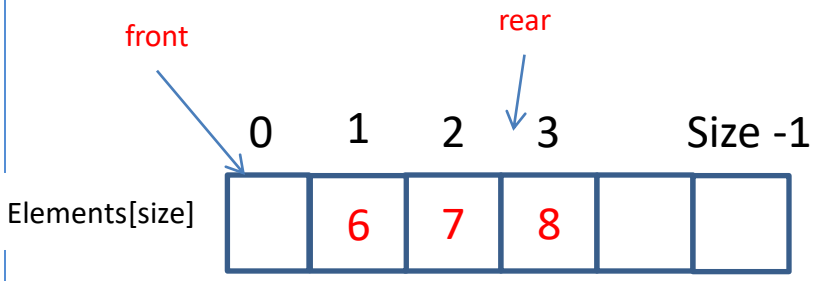
```
Push(stack_node   *top, stack_node  *x)
{x->next = top;
top = x;}
```

حذف عنصر از پشته(pop): حذف از ابتدا

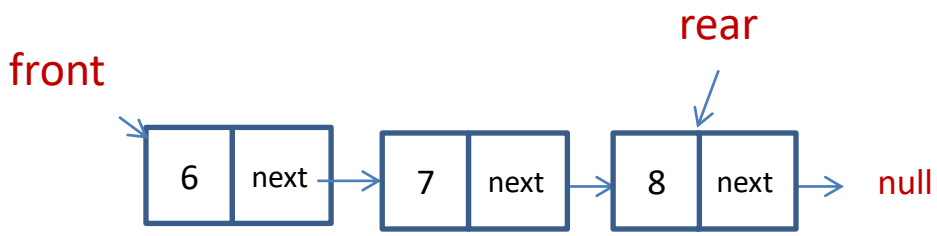
```
Pop(stack   *top)
{if (top !=null){
Stack_node  *temp = top;
top = top->next;
free(temp);}
```

پیاده سازی صف با استفاده از لیست پیوندی

- در پیاده سازی صف با آرایه داشتیم:

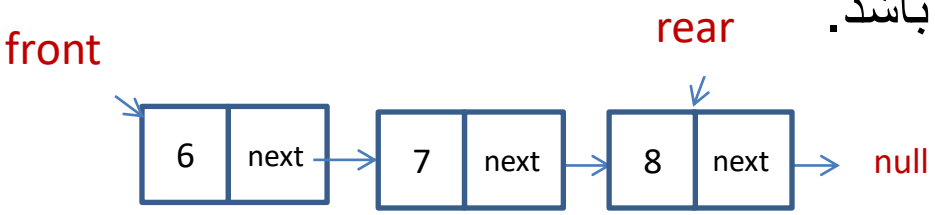


- حال اگر عناصر این صف را در بصورت غیر خطی با استفاده از لیست پیوندی پیاده سازی نماییم، داریم:



پیاده سازی صف با استفاده از لیست پیوندی

در پیاده سازی صف با لیست پیوندی، عملیات درج(Add_q) بایستی از انتهای صف و حذف(Remove_q) از ابتدای آن، انجام پذیرد تا خاصیت و رفتار صف مبنی خروج به ترتیب ورود برقرار باشد.

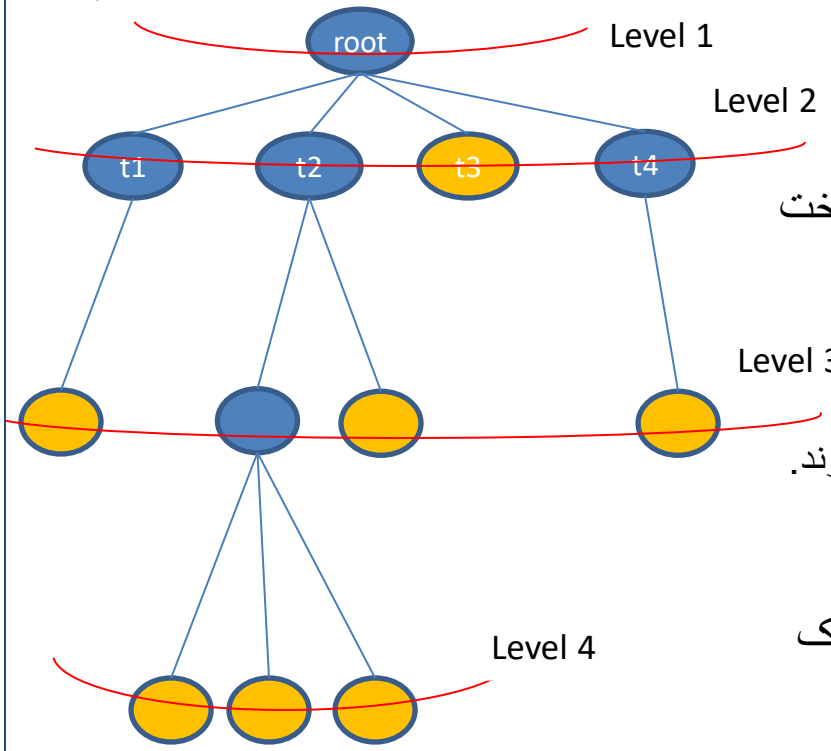


درج عنصر در صف(Add_q): درج در انتهای لیست

حذف عنصر از صف(Remove_q): حذف از ابتدای لیست

درخت(Tree)

- **تعریف**:یک درخت از یک گره خاص به نام ریشه و تعدادی از زیر درخت های...,t1,t2تشکیل شده که هر کدام خود یک درخت اند.(تعریف بازگشتی)



- **درجه هر گره**- تعداد زیردرخت های گره
- **درجه درخت**-بزرگترین درجه میان گره های درخت
- **گره برگ**- گره با درجه صفر
- سطح(level)

-اگرگره ی درسطح iباشد، فرزندان آن گره در سطح i+1قرار دارند.

- عمق(depth)- بیشترین سطح درخت
- همزادی(sibling)- گره های با گره پدر مشترک
- درخت mتایی

- درختی که بیشترین درجه میان گره های آن، m باشد.

درخت(Tree)

- انواع درخت:

درخت دودویی(binary tree)

درختی را دودویی(باینری) گویند که بیشترین درجه میان گره های آن 2 باشد. یعنی درجه گره های آن 0، 1 یا 2 است.

درخت عمومی(General Tree)

درختی را عمومی گویند هرگاه شامل حداقل یک گره با درجه بیشتر از 2 باشد.

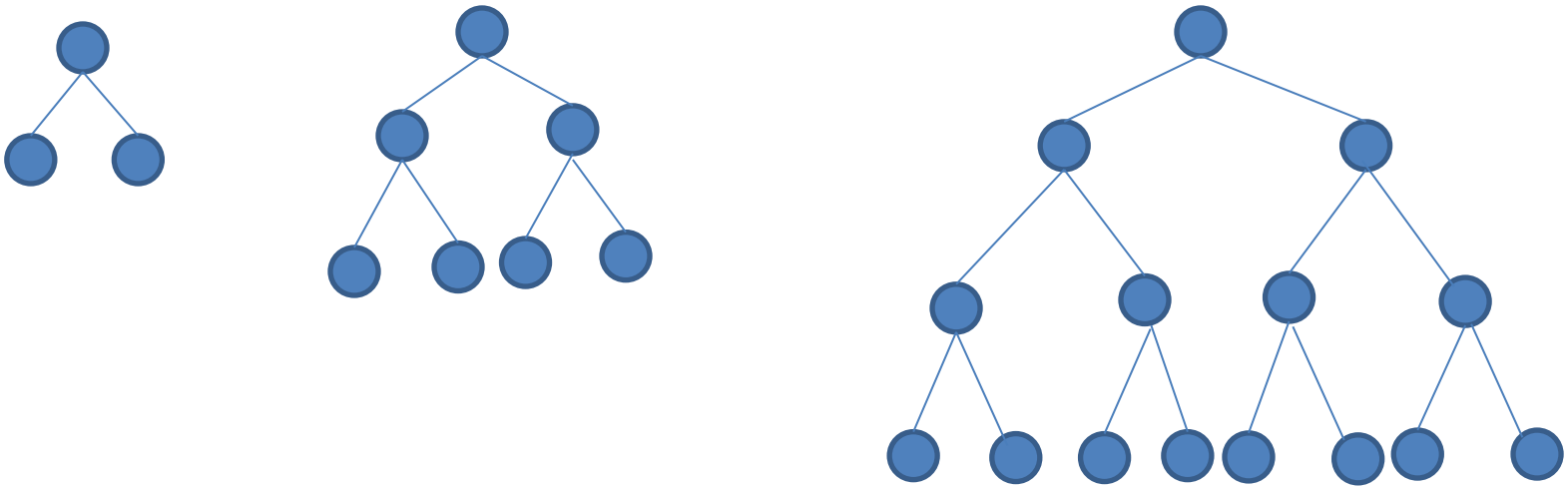
درخت دودویی(binary Tree)

- <u>درخت دودویی پر(Full Binary Tree)</u>

درخت باینری را پر گویند هرگاه:

الف)تمامی گره ها با درجه 2 باشند.

ب)تمامی گره های برگ در یک سطح قرار گرفته باشند.



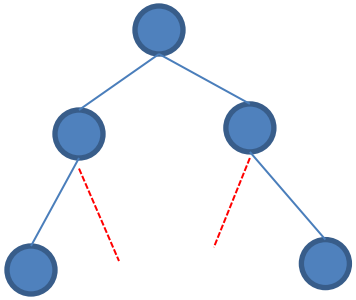
درخت دودویی(binary Tree)

- **درخت دودویی کامل(Complete Binary Tree)**

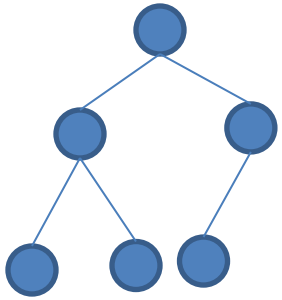
درخت باینری را کامل گویند هر گاه:

الف) تا عمق d – 1 درخت پر باشد.

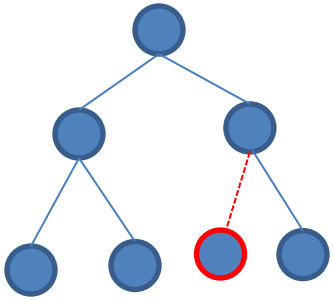
ب)در عمق d، برگها از سمت چپ در درخت به ترتیب اضافه شده باشند.



درخت کامل نیست



درخت کامل است



درخت کامل نیست

نکته: هر درخت پری، یک درخت کامل است ولی برعکس آن صحیح نیست.

چند قضیه مقدماتی در مورد درخت ها

- حداکثر تعداد گره های سطح i ازیک درخت دودویی برابر است با:

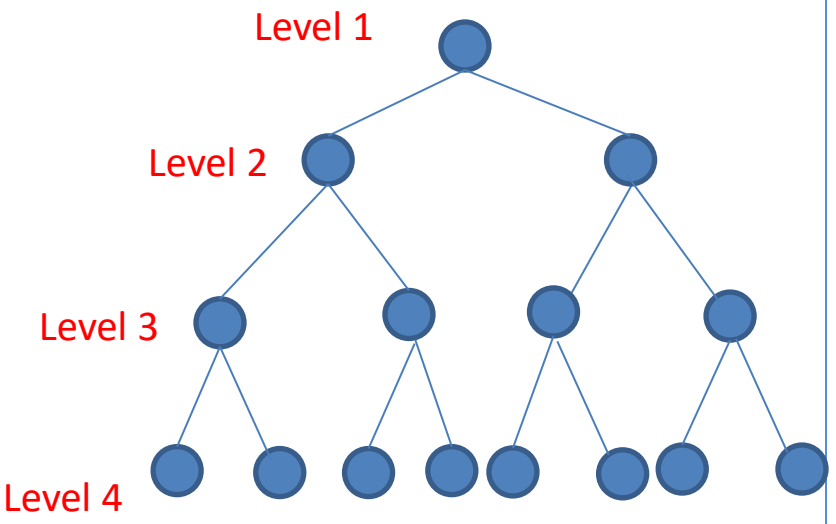
$$n = 2^{i-1}$$

i = 1
$n = 2^{1-1} = 2^0 = 1$

i = 2
$n = 2^{2-1} = 2^1 = 2$

i = 3
$n = 2^{3-1} = 2^2 = 4$

i = 4
$n = 2^{4-1} = 2^3 = 8$

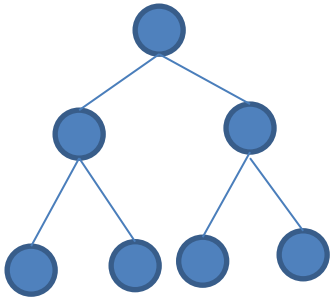


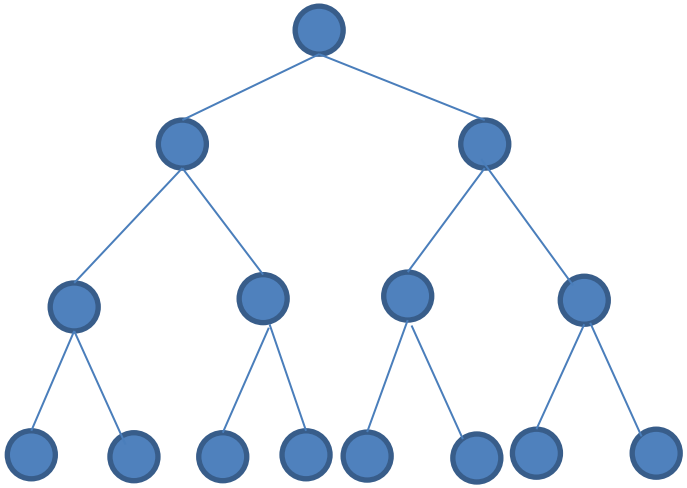
چند قضیه مقدماتی در مورد درخت ها

- حداکثر تعداد گره های یک درخت دودویی با عمق k برابر است با:

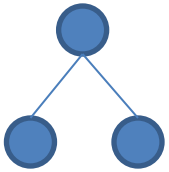
$n = 2^k - 1$

K = 3
$2^k - 1 = 7$
n = 7





K = 4
$2^k - 1 = 15$
n = 15



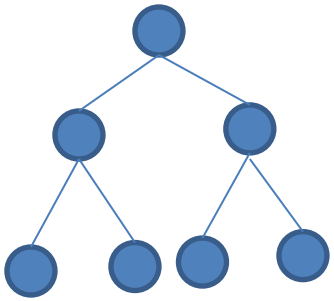
K = 2
$2^k - 1 = 3$
n = 3

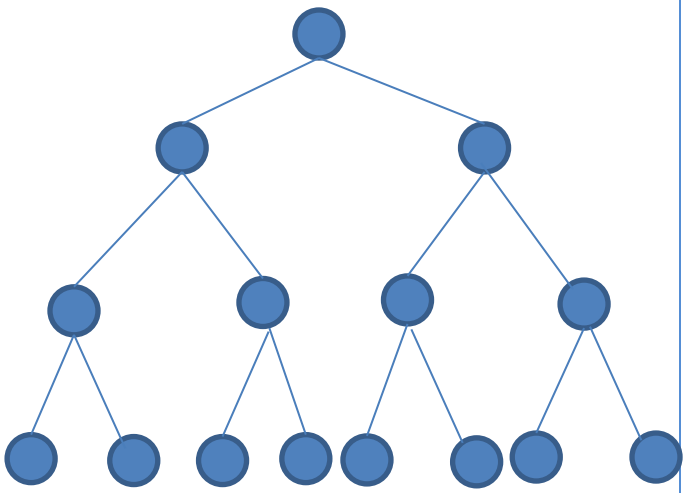
چند قضیه مقدماتی در مورد درخت ها

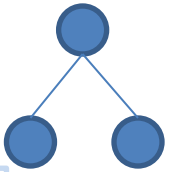
- عمق یک درخت دودویی پر با n گره برابر است با:

$$k = \log(n+1)$$

K = log(7+1)
2^k= 8
K = 3







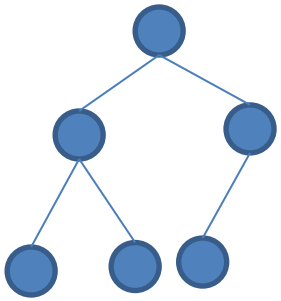
K = log(3+1)
2^k= 4
K = 2

K = log(15+1)
2^k= 16
K = 4

چند قضیه مقدماتی در مورد درخت ها

- عمق یک درخت دودویی کامل با n گره برابر است با:

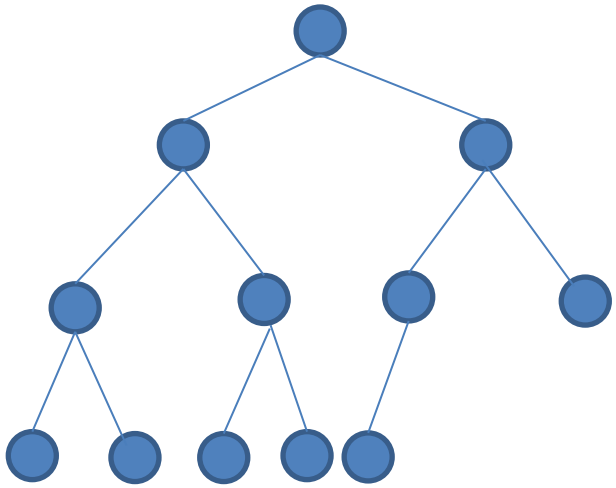
$$k = \lceil \log(n+1) \rceil$$



K = log(6+1)
Log(4)<log(7)<log(8)
2 < ? < 3

K = ⌈log (6+1)⌉

K = 3



K = log(12+1)
Log(8)<log(13)<log(16)
3 < ? < 4

چند قضیه مقدماتی در مورد درخت ها

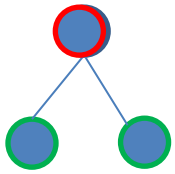
- در درخت های دودویی همواره داریم:

$n_0 = n_2 + 1$

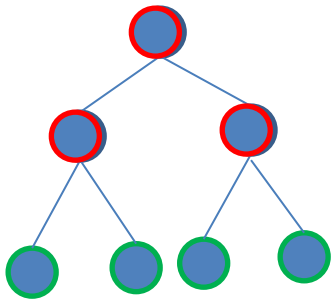
n_0: تعداد گره های با درجه 0(برگ)

n_2: تعداد گره های با درجه 2

$n_0 = 2$
$n_2 = 1$



$n_0 = 4$
$n_2 = 3$



$n_0 = 6$
$n_2 = 5$

