ساختمان داده ها

درخت دودویی اندیس دهی شده

(Indexed Binary Tree)

(Order Statistic Tree)

مدرس: غیاثی‌شیرازی

دانشگاه فردوسی مشهد

یادآوری نوع داده مجرد (ADT) لیست خطی

- Get(index)
- Insert(index, value)
- Delete(index)

L = 's', 'a', 'a', 'm'

0 1 2 3

Get(3) → 'm'

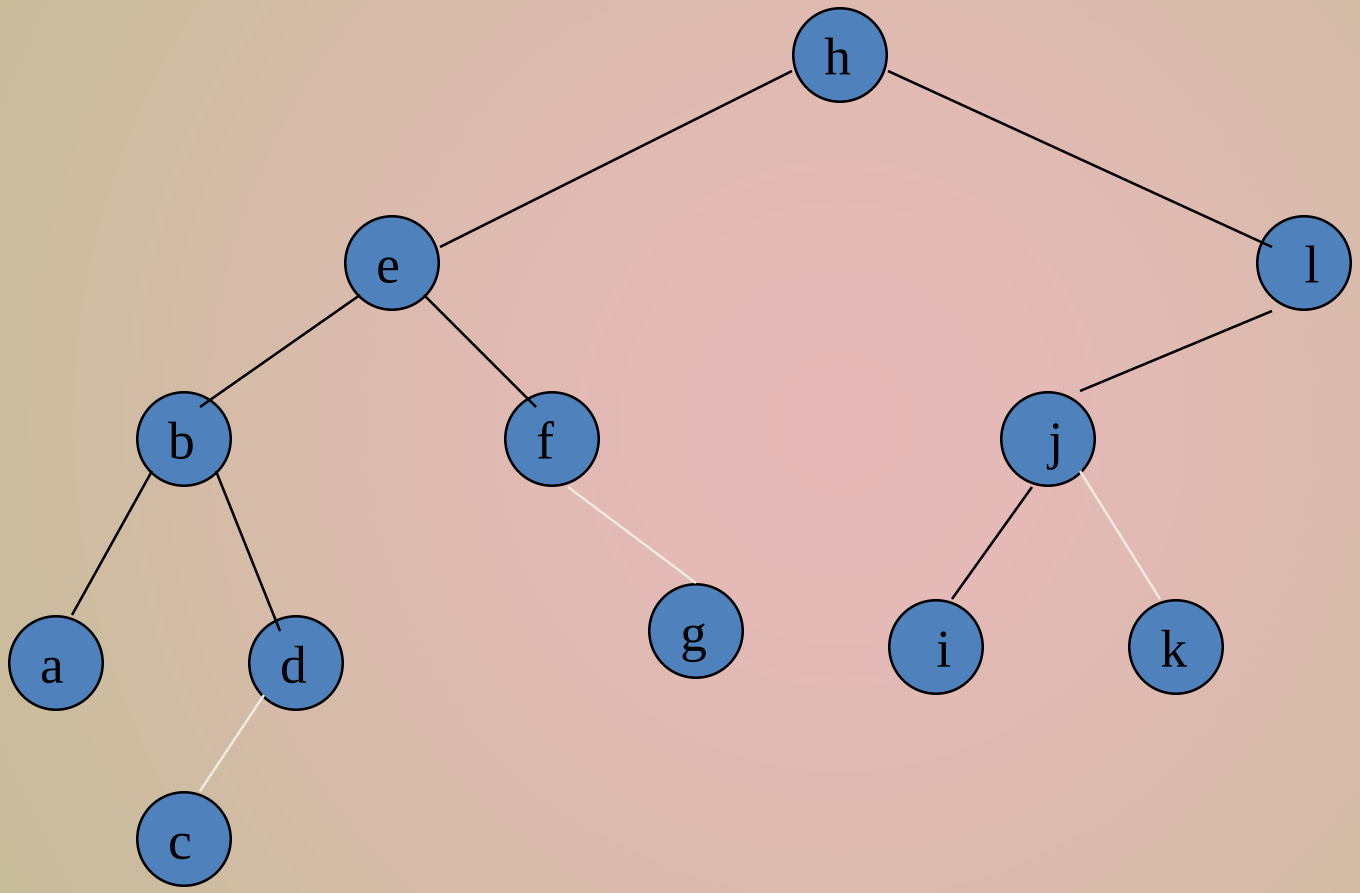
insert(2, 'l') → L = 's', 'a', 'l', 'a', 'm'

Delete(2) → L = 's', 'a', 'm'

Get	Insert	Delete	
$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	Array Linear List
$\Theta(n)$	Get $\Theta(n) + \Theta(1)$	Get $\Theta(n) + \Theta(1)$	Linked Linear List
$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(\log n)$	[Balanced] Indexed BinaryTree

Inorder پیمایش

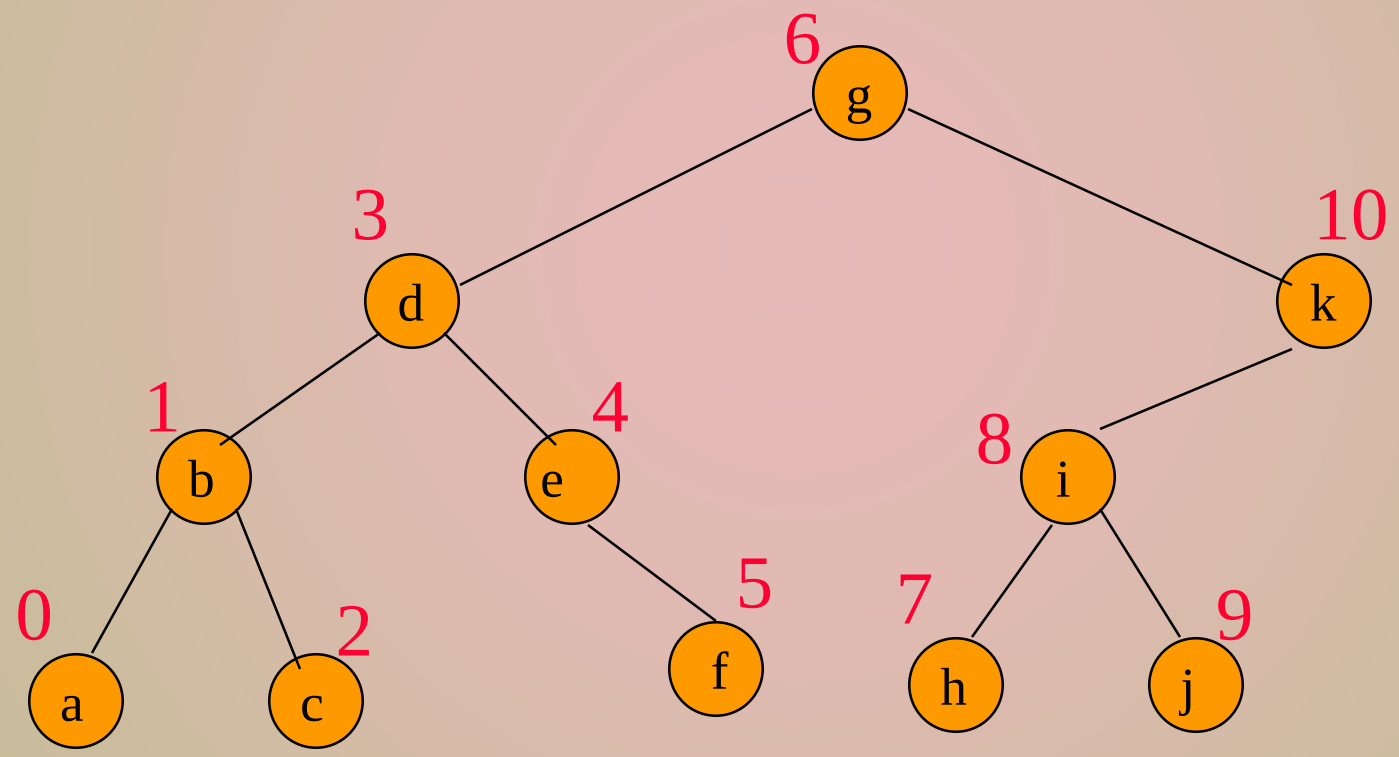
ایجاد تناظر بین درخت دودویی و لیست خطی



list = [a,b,c,d,e,f,g,h,i,j,k,l]

بررسی ایده ذخیره اندیس هر گره

Get O(ارتفاع درخت)



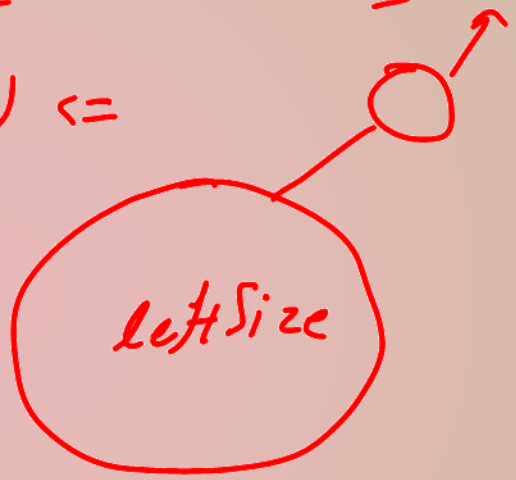
O(n)

Delete و insert

ایده صحیح: ذخیره اندیس هر گره در لیست خطی متناظر با زیر درخت آن گره

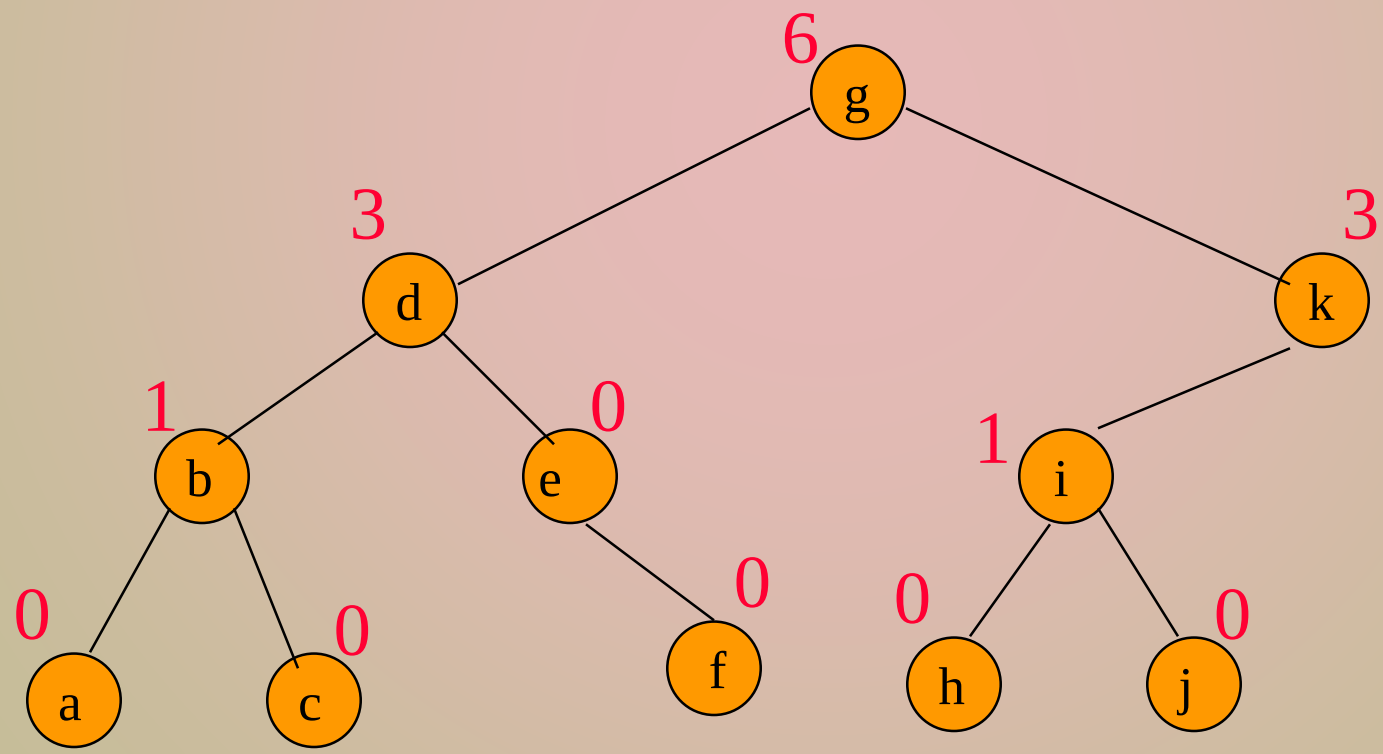
leftSize+1 این عضو در لیست مبنا ترتیب زیر لیست خودش است

=> اندیس آن برابر leftSize است.



روش دسترسی سریع به گره i ام در پیمایش میانوند

- هر گره دارای متغیری به نام leftSize است که تعداد گره‌ها را در زیردرخت سمت چپ گره نشان می دهد.



روش دسترسی سریع بر اساس اندیس

Get

- توجه کنید که عناصر از صفر اندیس دهی شده اند.

- if index = x.leftSize desired element is …….
- if index < x.leftSize desired element is …….
- if index > x.leftSize desired element is …….

Get(index)

روش دسترسی سریع بر اساس اندیس

- توجه کنید که عناصر از صفر اندیس دهی شده اند.

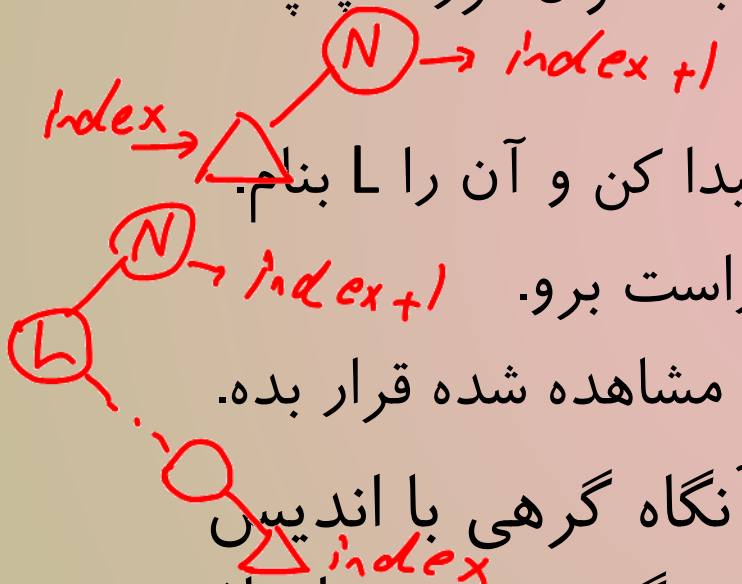
- if index = x.leftSize desired element is x.element
- if index < x.leftSize desired element is index'th element in left subtree of x
- if index > x.leftSize desired element is (index - x.leftSize-1)'th element in right subtree of x

درج (Insert)

- عمل درج باید به نحوی انجام شود که پس از درج، عنصر مورد نظر در محل صحیح در پیمایش میان ترتیب قرار گیرد.
- ترجیح ما این است که گرهی که اضافه می شود به انتهای درخت اضافه شود تا پیاده سازی ساده تری داشته باشد.
- درج یک الگوریتم ندارد. الگوریتم هایی که بیان می شود خواص مورد نظر را دارد.

الگوریتم اول برای درج با دستور insert(index,value)

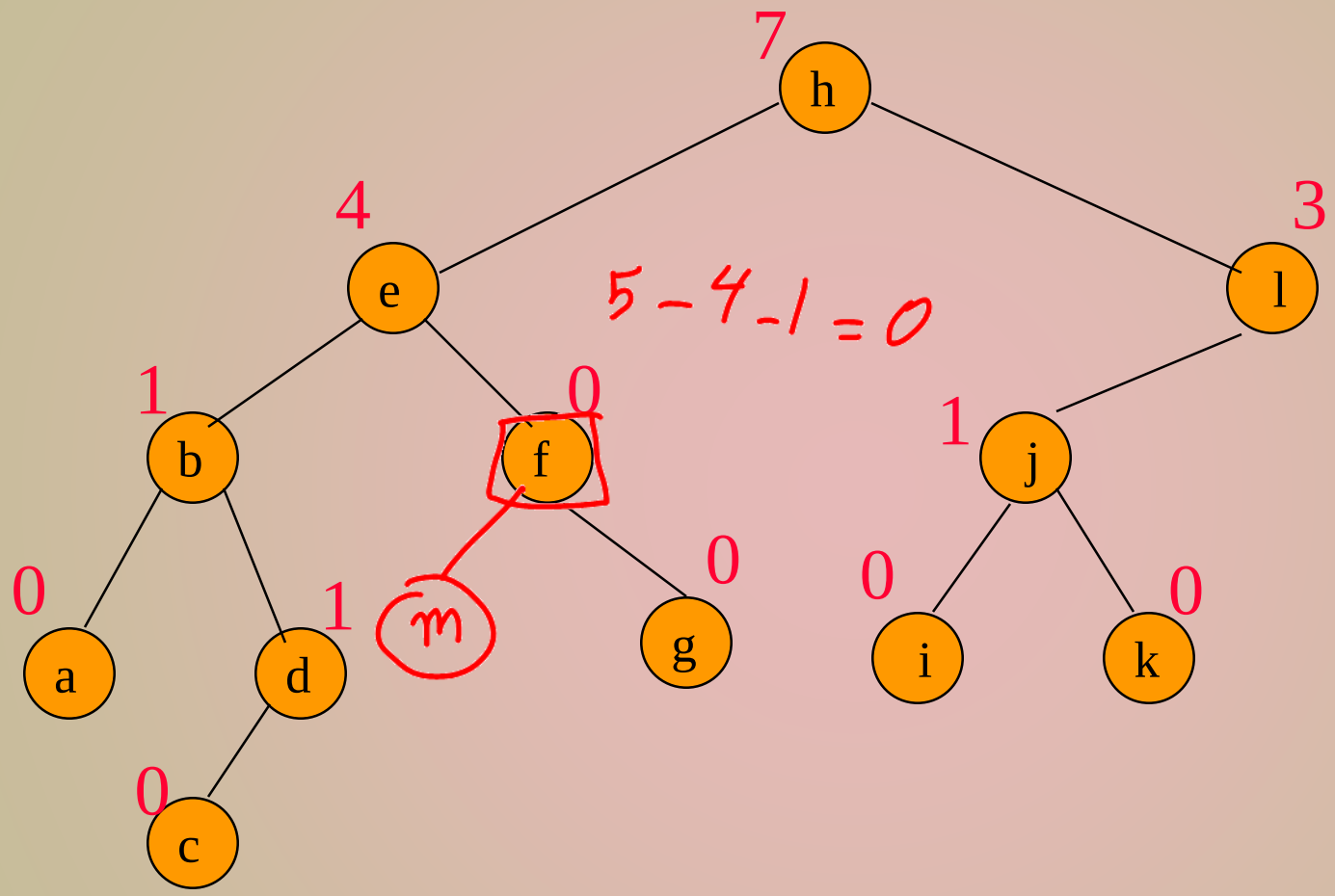
- فرض می کنیم گره N دارای اندیس مورد نظر است.
  - اگر N فرزند چپ ندارد، گره جدید را به عنوان فرزند چپ N اضافه کن.
  - در غیر این صورت فرزند چپ N را پیدا کن و آن را L بنام.
  - با شروع از L تا آنجا که می توانی به راست برو.
  - گره جدید را فرزند راست آخرین گره مشاهده شده قرار بده.
- اگر index برابر طول لیست باشد، آنگاه گرهی با اندیس index وجود ندارد. در این حالت باید گره جدیدی اضافه کنیم که در ترتیب میانوند بعد از همه گره ها باشد.



در صورت وجود گره های ساختگی

- در صورتی که درخت دودویی دارای InorderEnd باشد، آنگاه حتی اگر بخواهیم گرهی را به انتهای لیست اضافه کنیم، باز هم همان الگوریتم قبلی کار می کند.
- اگر index برابر طول لیست باشد، آنگاه گره InorderEnd دارای اندیس index است.

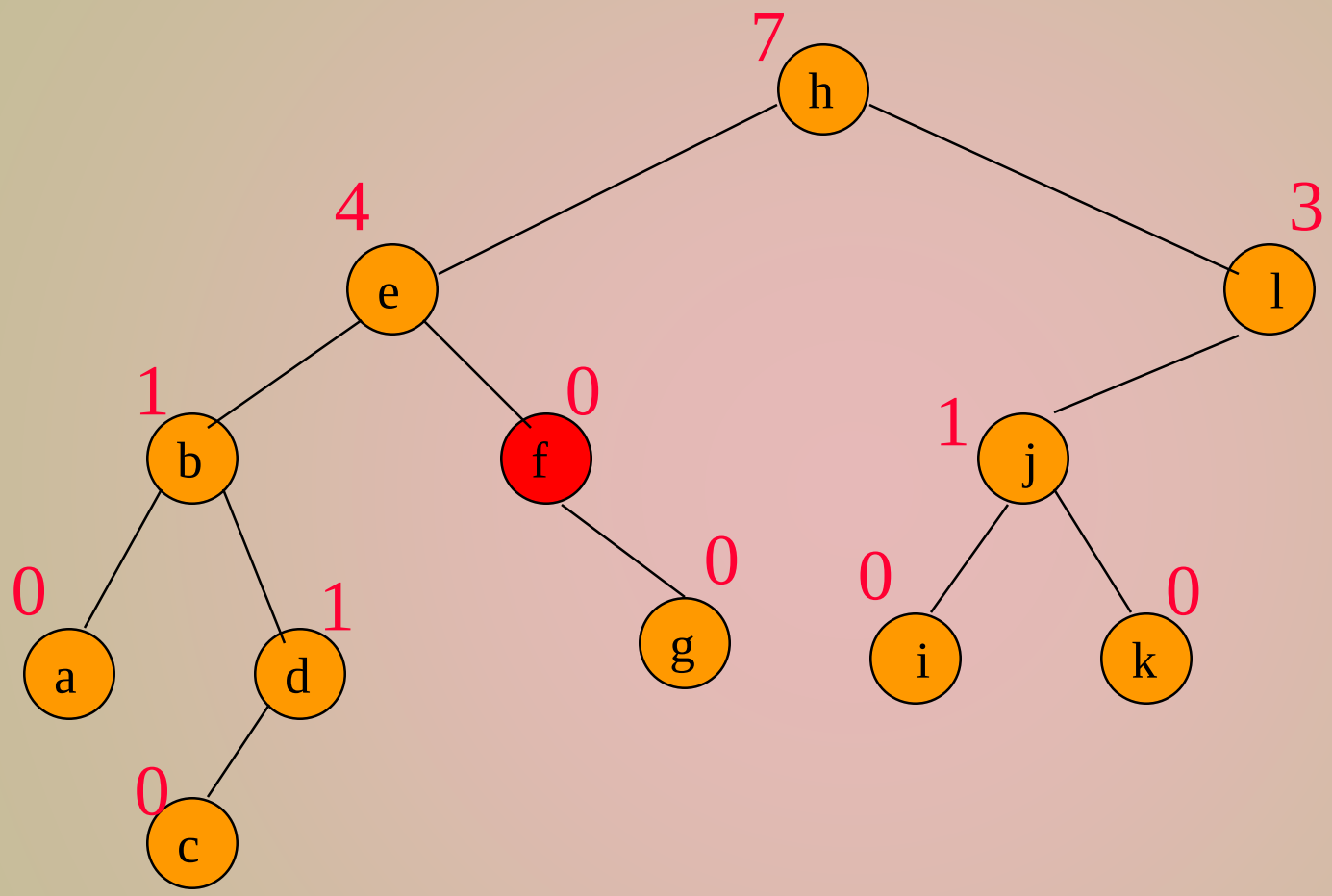
مثال اول: Insert(5,’m’)



list = [a,b,c,d,e,f,g,h,i,j,k,l]

لیست مرجع [a, b, c, d, e, m, f, g, h, i, j, k, l]

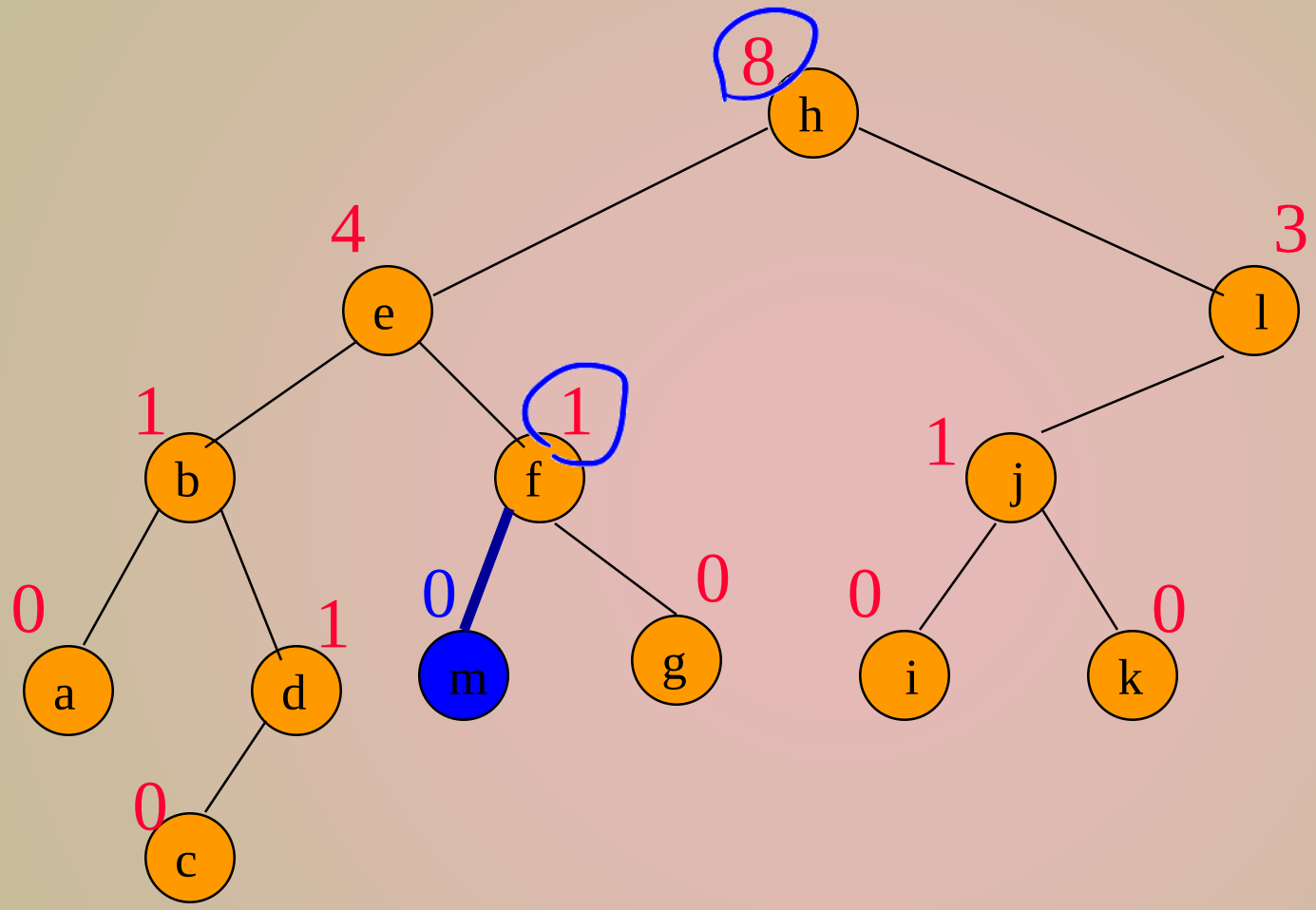
Insert(5,'m')



list = [a,b,c,d,e, m,f,g,h,i,j,k,l]

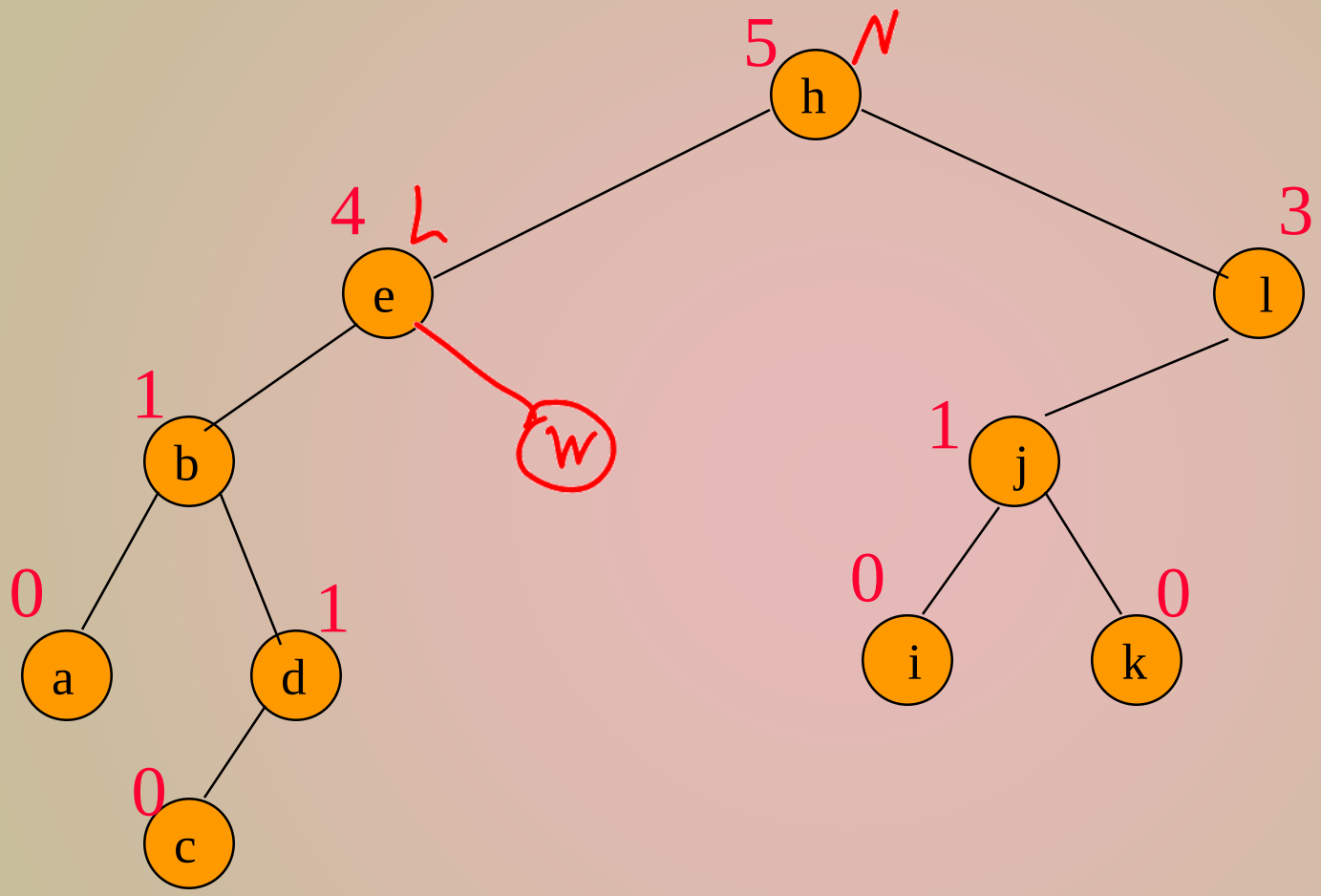
find node with index 5 (f)

Insert(5,'m')



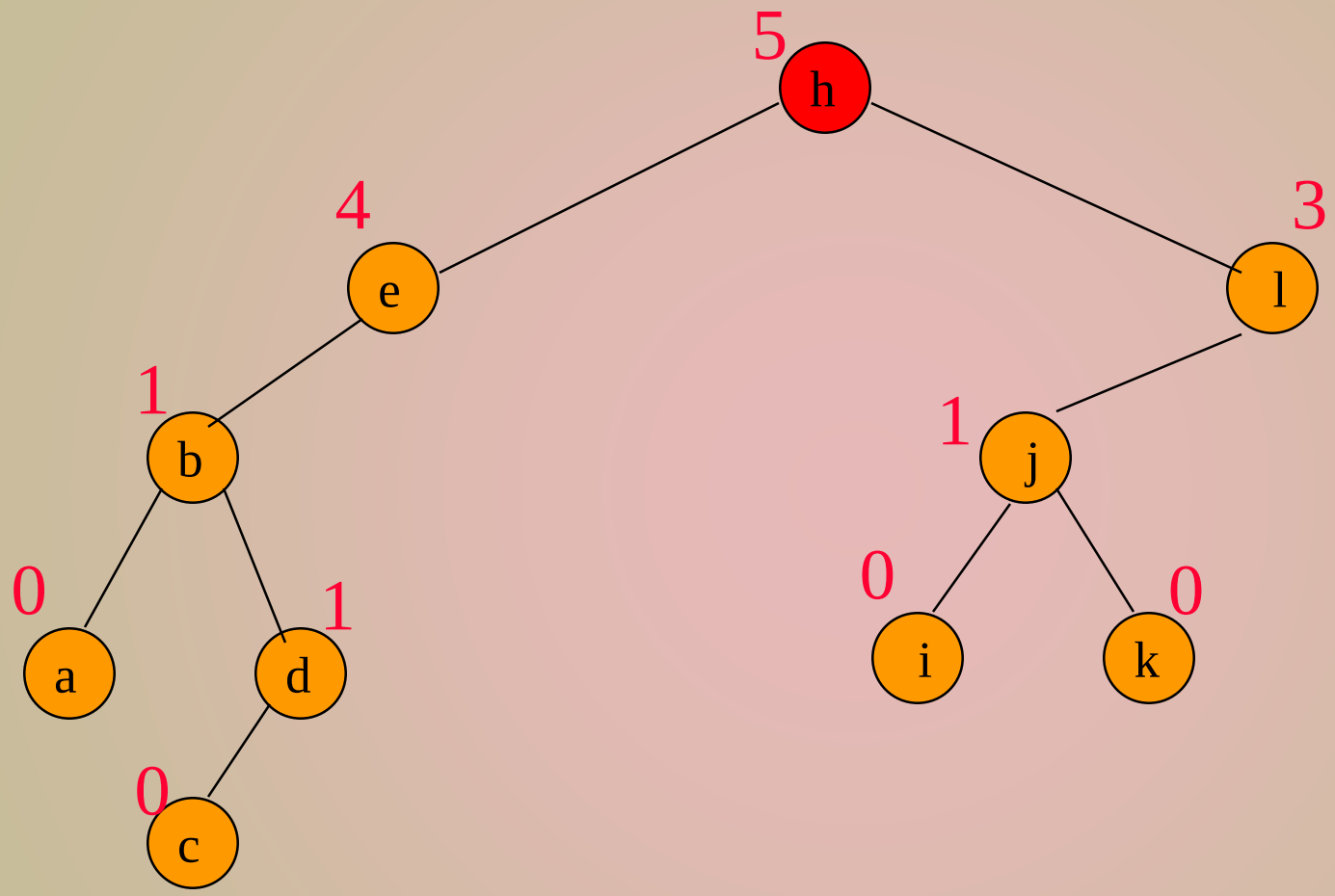
add m as left node of f

مثال دوم: Insert(5,’w’)



list = [a,b,c,d,e,h,i,j,k,l]

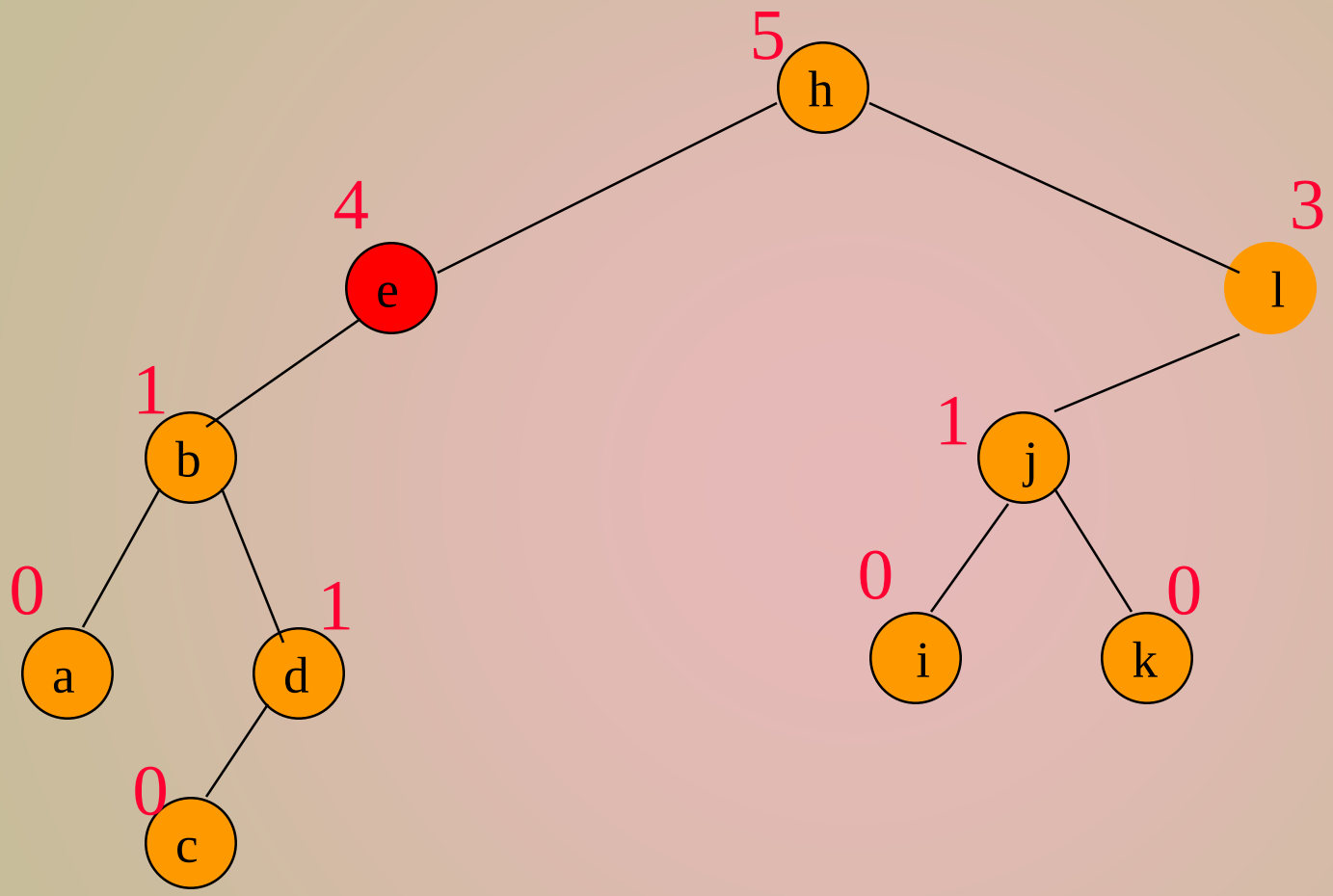
Insert(5,'w')



list = [a,b,c,d,e, w, h,i,j,k,l]

find node with index 5 (h)

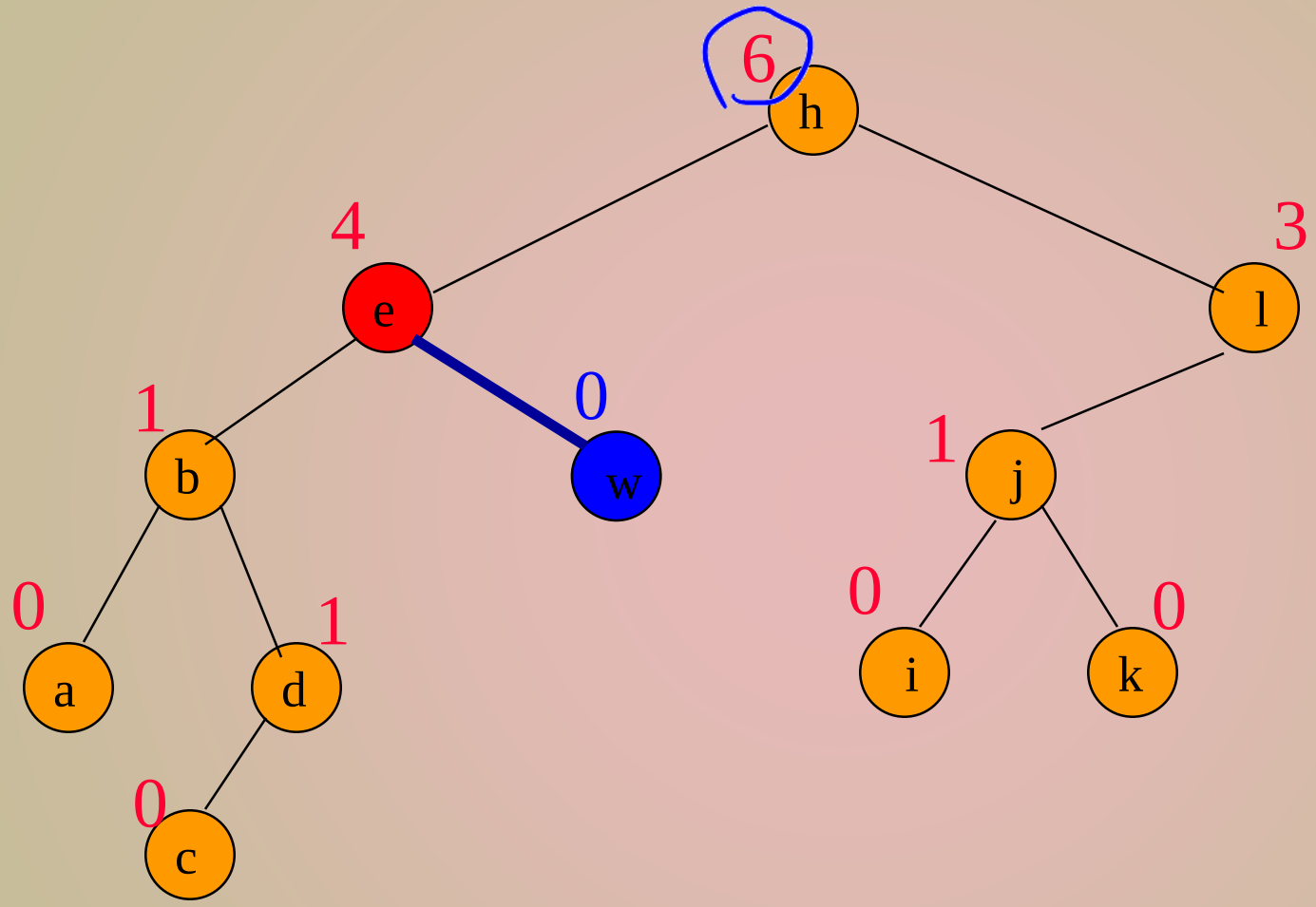
Insert(5,'w')



list = [a,b,c,d,e, w, h,i,j,k,l]

Find left child of h (e)

Insert(5,'w')

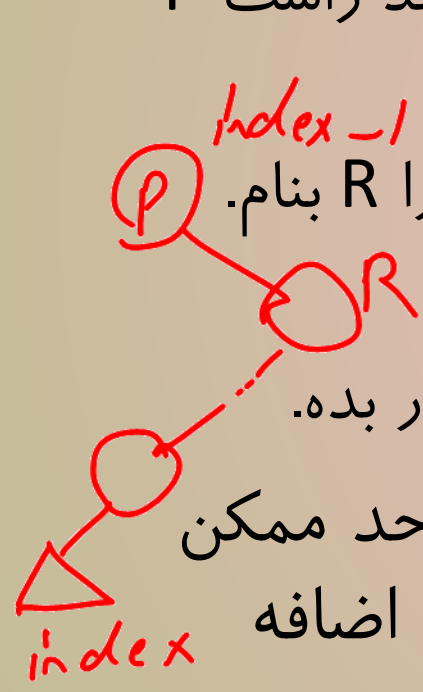


list = [a,b,c,d,e, w, h,i,j,k,l]

Add w as the rightmost child of e.

الگوریتم دوم برای درج با دستور insert(index,value)

- فرض می کنیم گره P دارای اندیس index-1 است.
  - اگر P فرزند راست ندارد، گره جدید را به عنوان فرزند راست P اضافه کن.
  - در غیر این صورت فرزند راست P را پیدا کن و آن را R بنام.
  - با شروع از R تا آنجا که می توانی به چپ برو.
  - گره جدید را فرزند چپ آخرین گره مشاهده شده قرار بده.
- اگر index برابر صفر باشد آنگاه باید از ریشه تا حد ممکن به چپ برویم و گره جدید را به عنوان فرزند چپ اضافه کنیم.

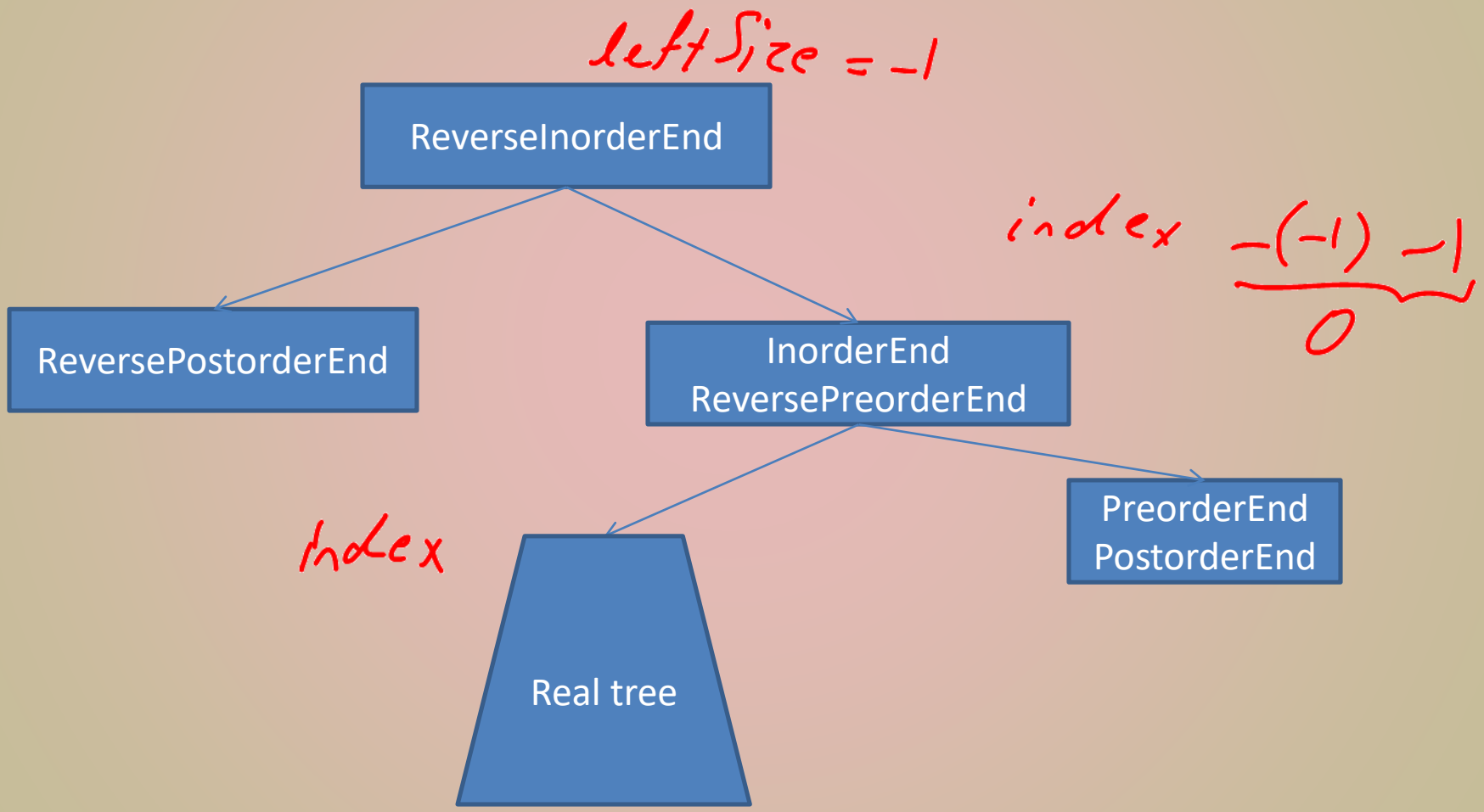


حالت خاص

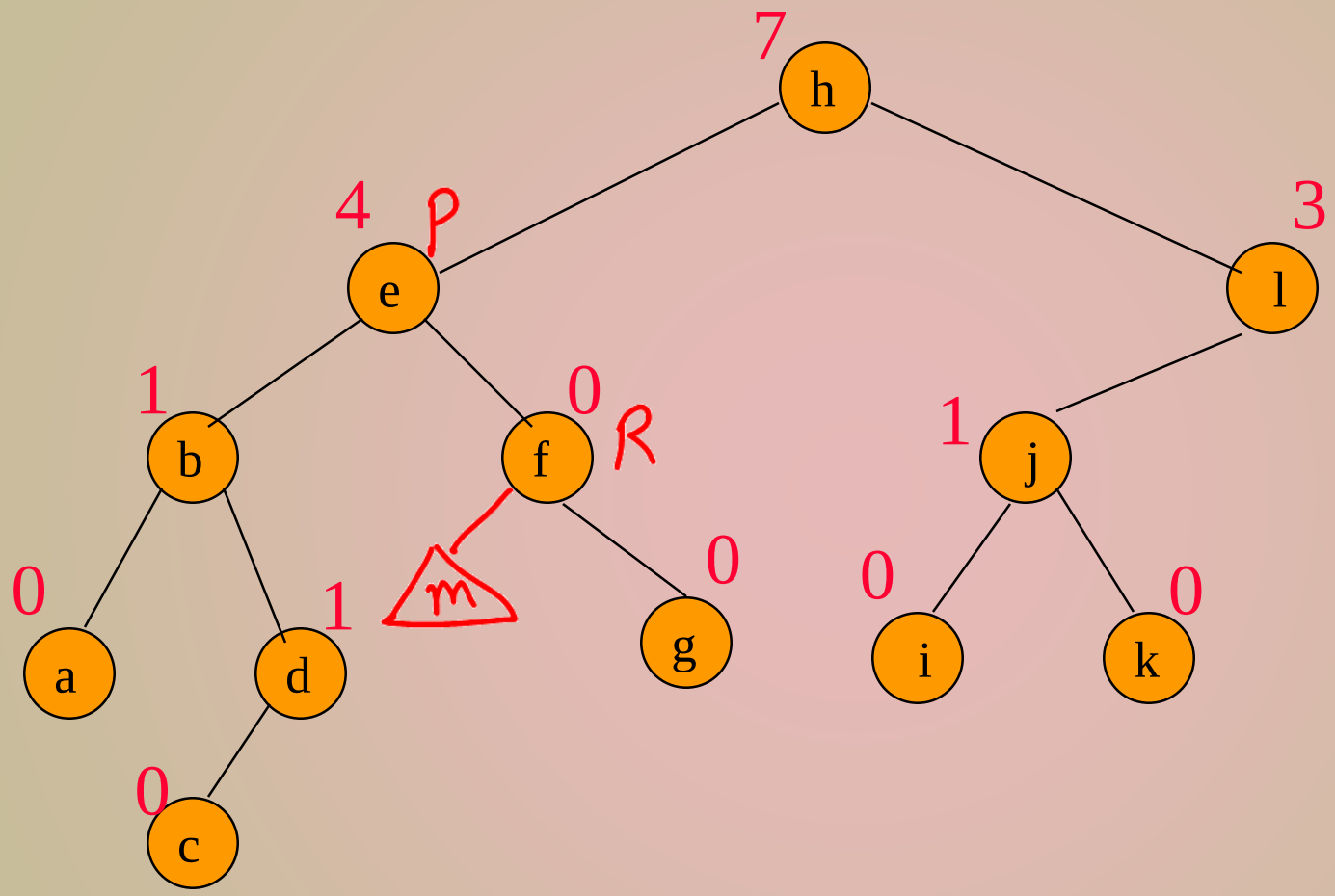
در صورت وجود گره های ساختگی

- در صورتی که درخت دودویی دارای RevInorderEnd باشد، آنگاه حتی اگر بخواهیم گرهی را به ابتدای لیست اضافه کنیم، باز هم همان الگوریتم قبلی کار می کند.
- اگر index برابر 0 باشد، آنگاه گره RevInorderEnd دارای اندیس 1- است.
  - بنابراین باید متغیر leftSize برای این گره برابر 1- باشد.

یادآوری ساختار کلی درخت دودویی با چند گره ساختگی

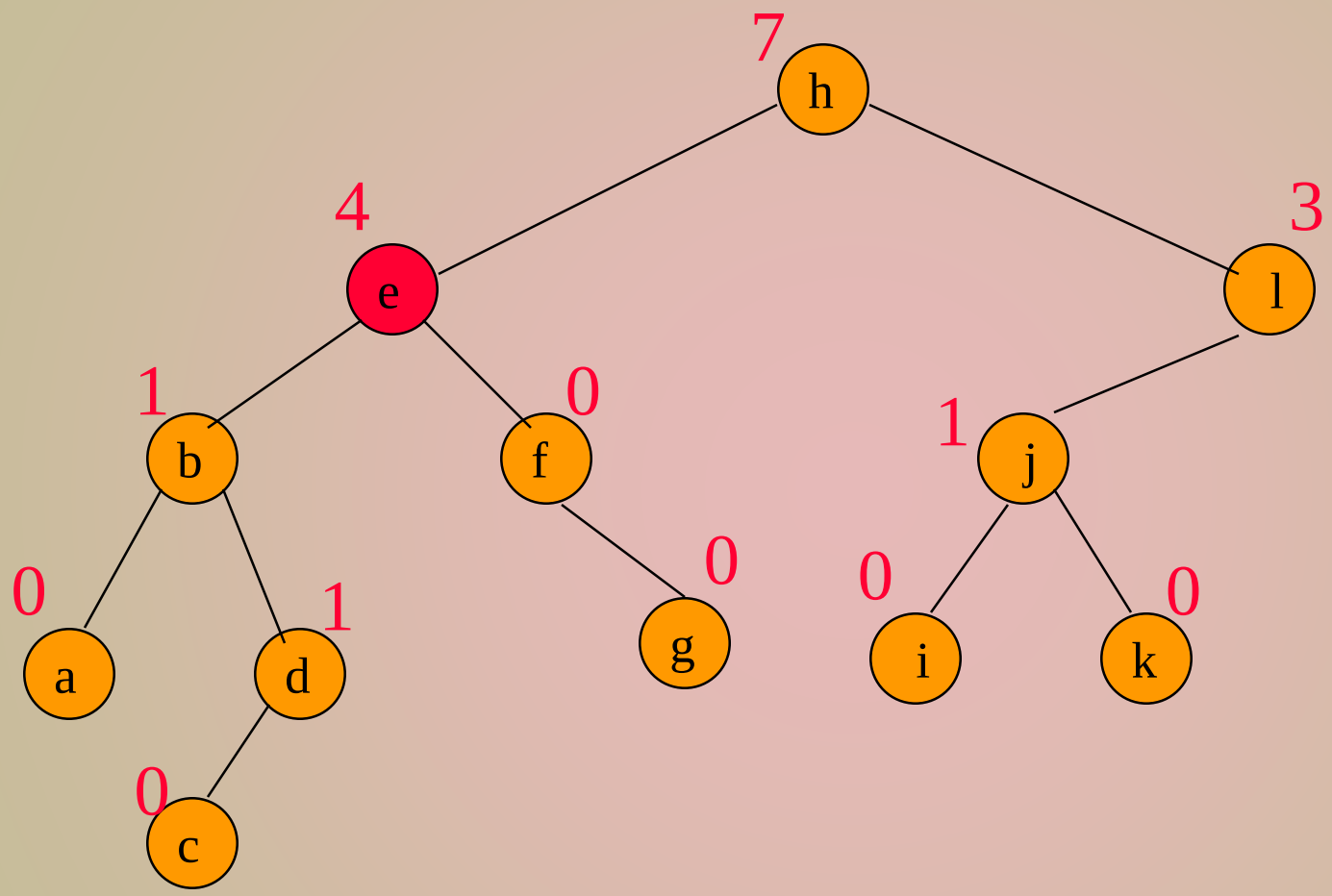


مثال اول: Insert(5,’m’)



list = [a,b,c,d,e,f,g,h,i,j,k,l]

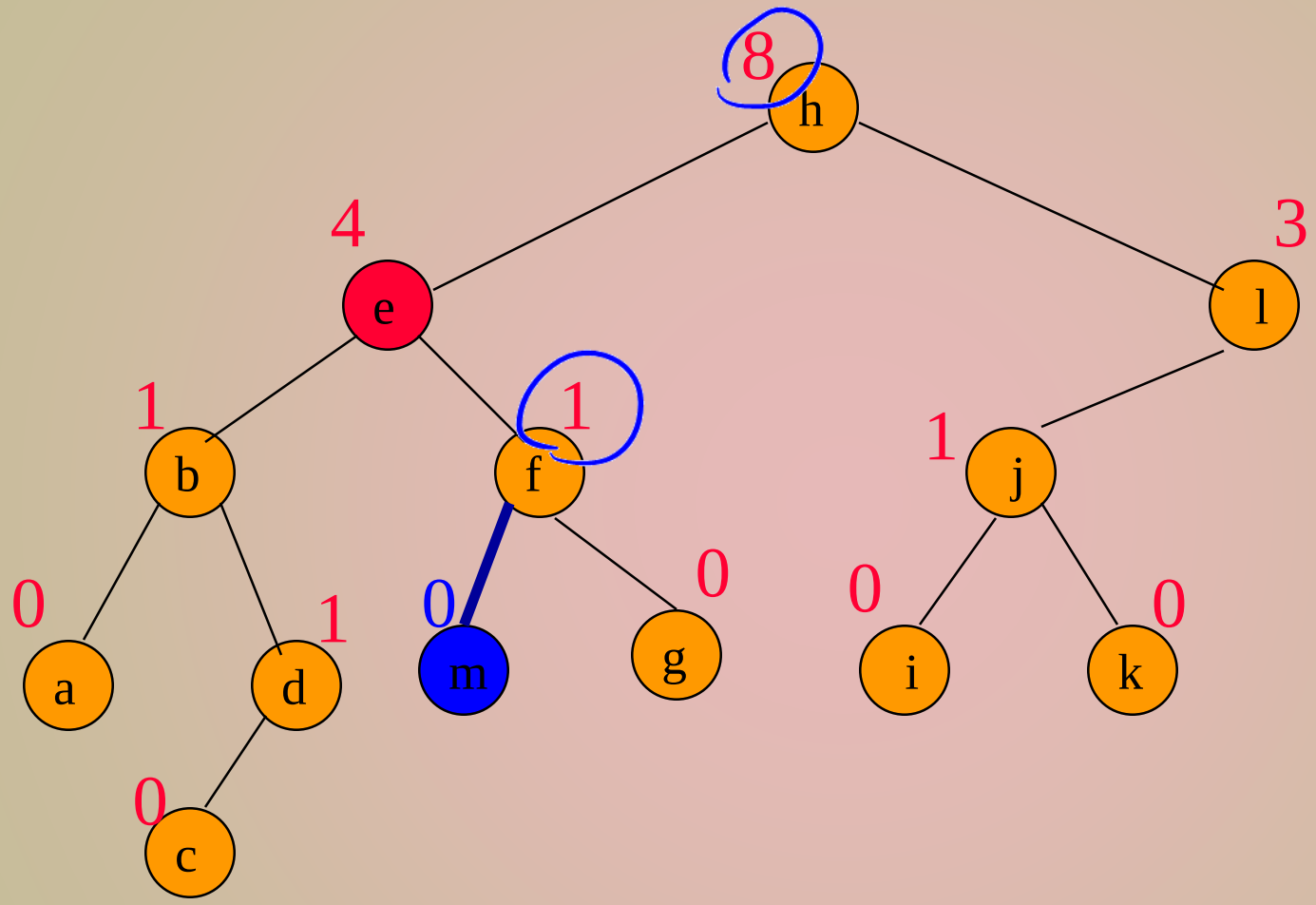
Insert(5,’m’)



list = [a,b,c,d,e, m,f,g,h,i,j,k,l]

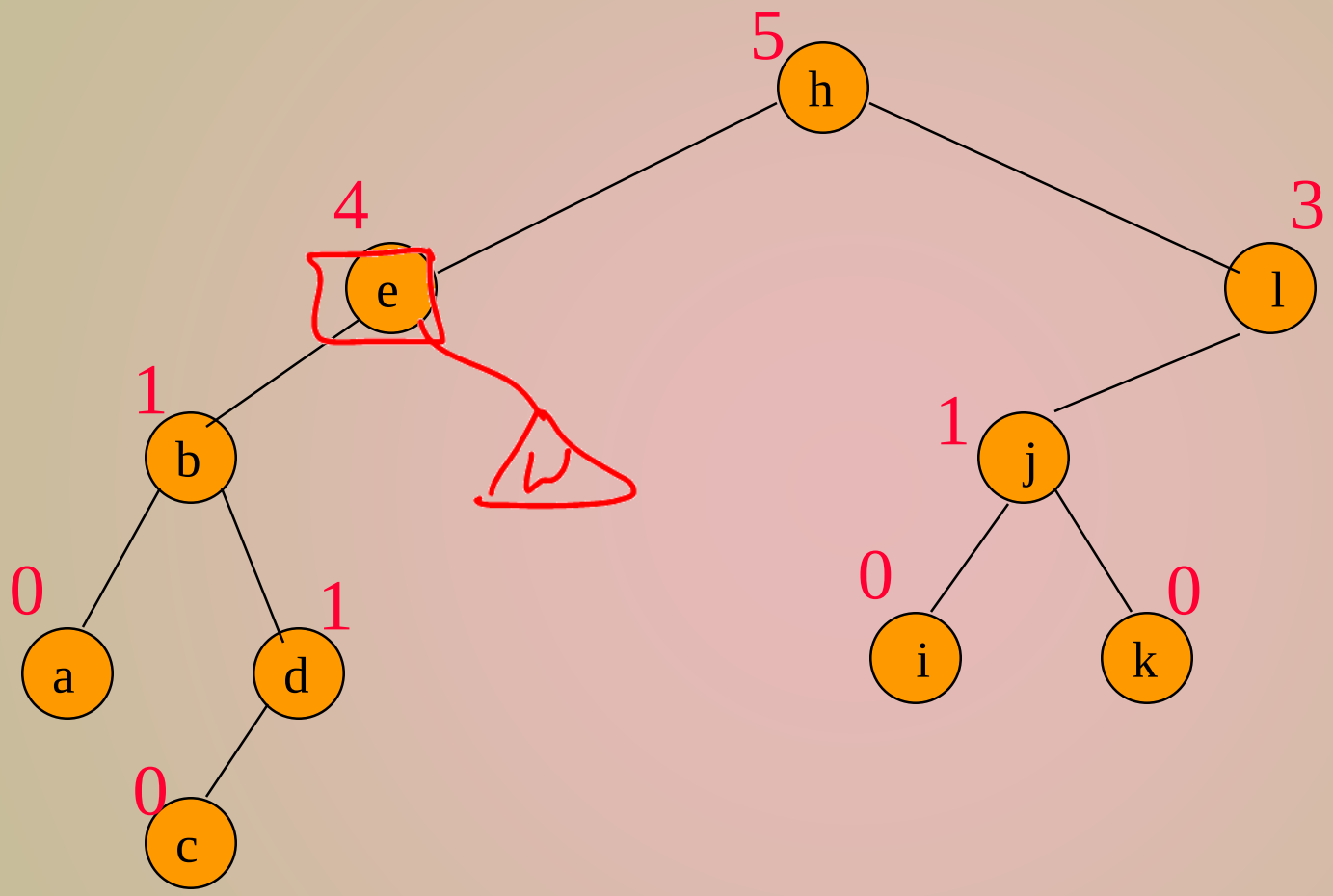
find node with element 4 (e)

Insert(5,'m')



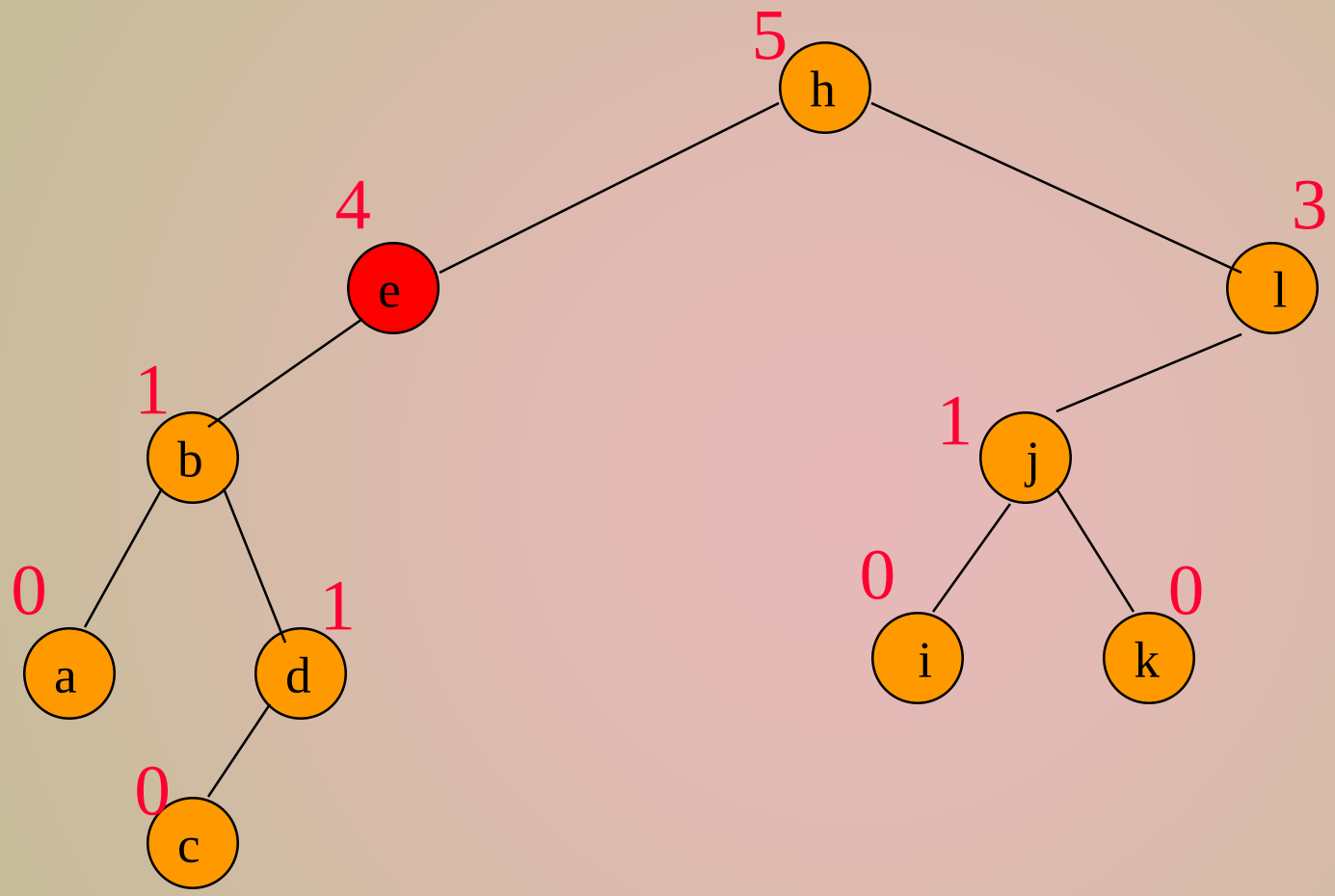
add m as leftmost node in right subtree of e

مثال دوم: Insert(5,’w’)



list = [a,b,c,d,e,h,i,j,k,l]

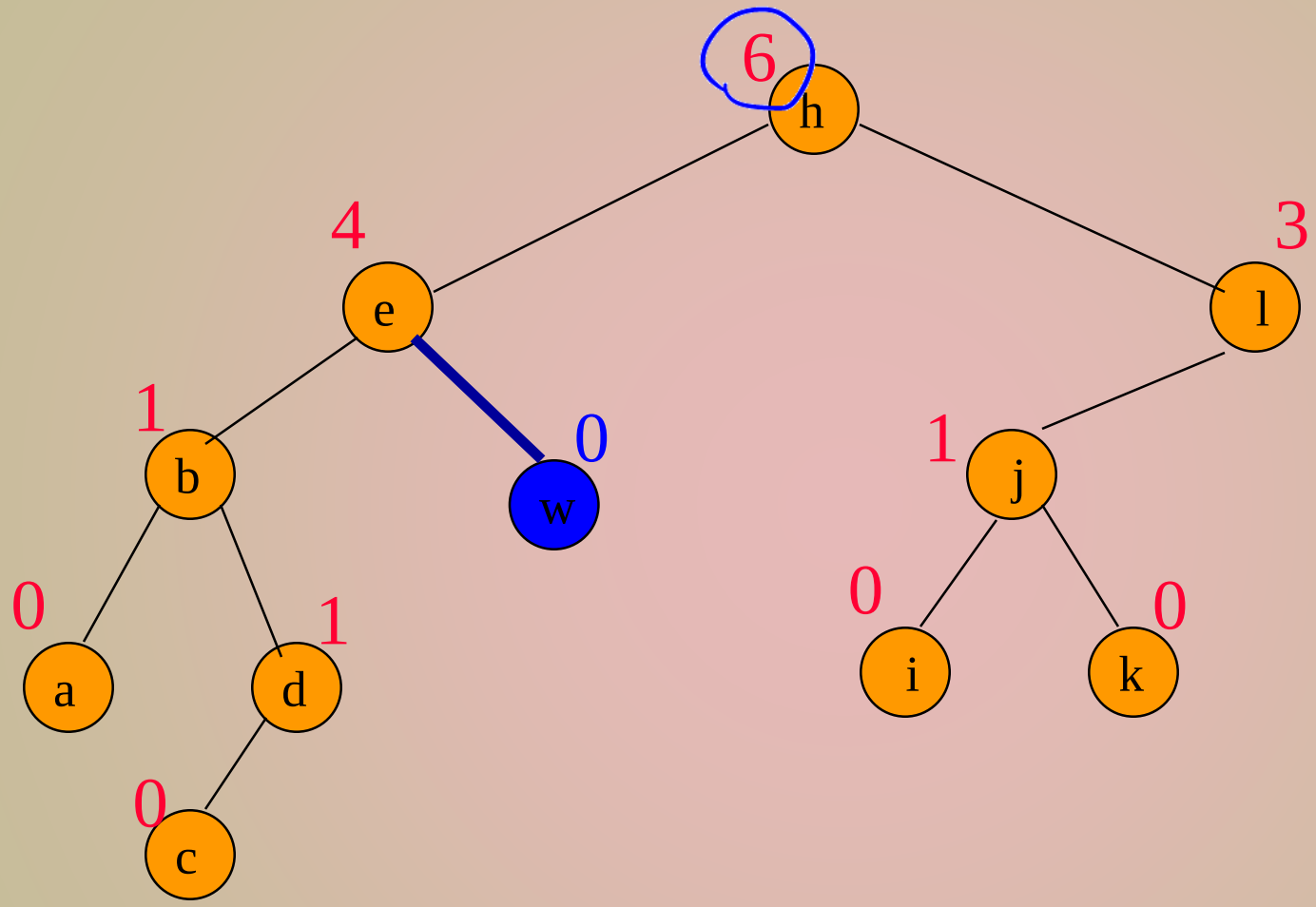
Insert(5,’w’)



list = [a,b,c,d,e, w, h,i,j,k,l]

find node with element 4 (e)

Insert(5,'w')



list = [a,b,c,d,e, w, h,i,j,k,l]

Add w as the right node of e

الگوریتم دستور حذف: Delete(index)

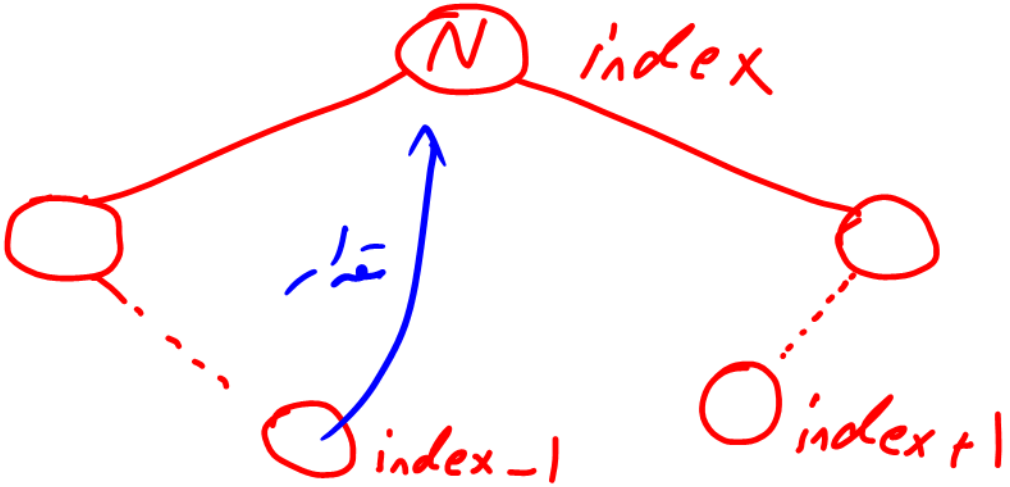
- فرض می کنیم گره N دارای اندیس مورد نظر است.
  - اگر N یک گره برگ است .....
  - اگر N گرهی با درجه ۱ است .....
  - اگر N دارای دو فرزند باشد .....

الگوریتم دستور حذف: Delete(index)

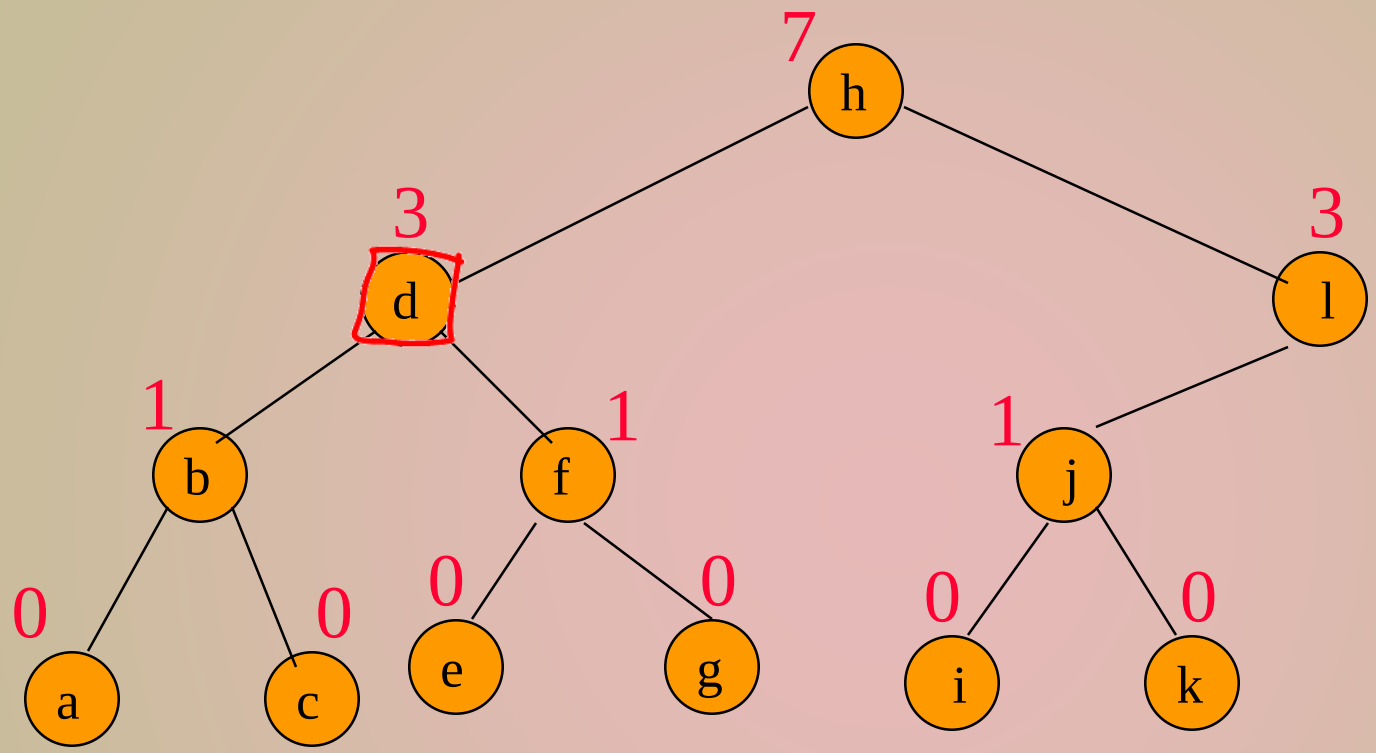
- فرض می کنیم گره N دارای اندیس مورد نظر است.
  - اگر N یک گره برگ است کافی است گره N را حذف کنیم.
  - اگر N گرهی با درجه ۱ است، کافی است آن را حذف کنیم و فرزندش را فرزند پدر N قرار دهیم.
  - اگر N دارای دو فرزند باشد مساله دشوارتر است.

حذف گرهی با درجه ۲

- ایده کلی: پس از حذف N گرهی با مقدار مناسب را جایگزین آن کنیم. دو کاندید داریم:
  - گرهی از زیردرخت چپ که بعد از همه مشاهده می شود.
  - گرهی از زیردرخت راست که قبل از همه مشاهده می شود.
- نکته: گره های فوق هیچگاه از درجه ۲ نیستند.
  - زیرا در این صورت فرزند چپ آنها قبل از آنها و فرزند راست آنها بعد از آنها مشاهده می شود.
- بنابراین کافی است یکی از دو گره گفته شده را حذف کنیم و جایگزین N کنیم.

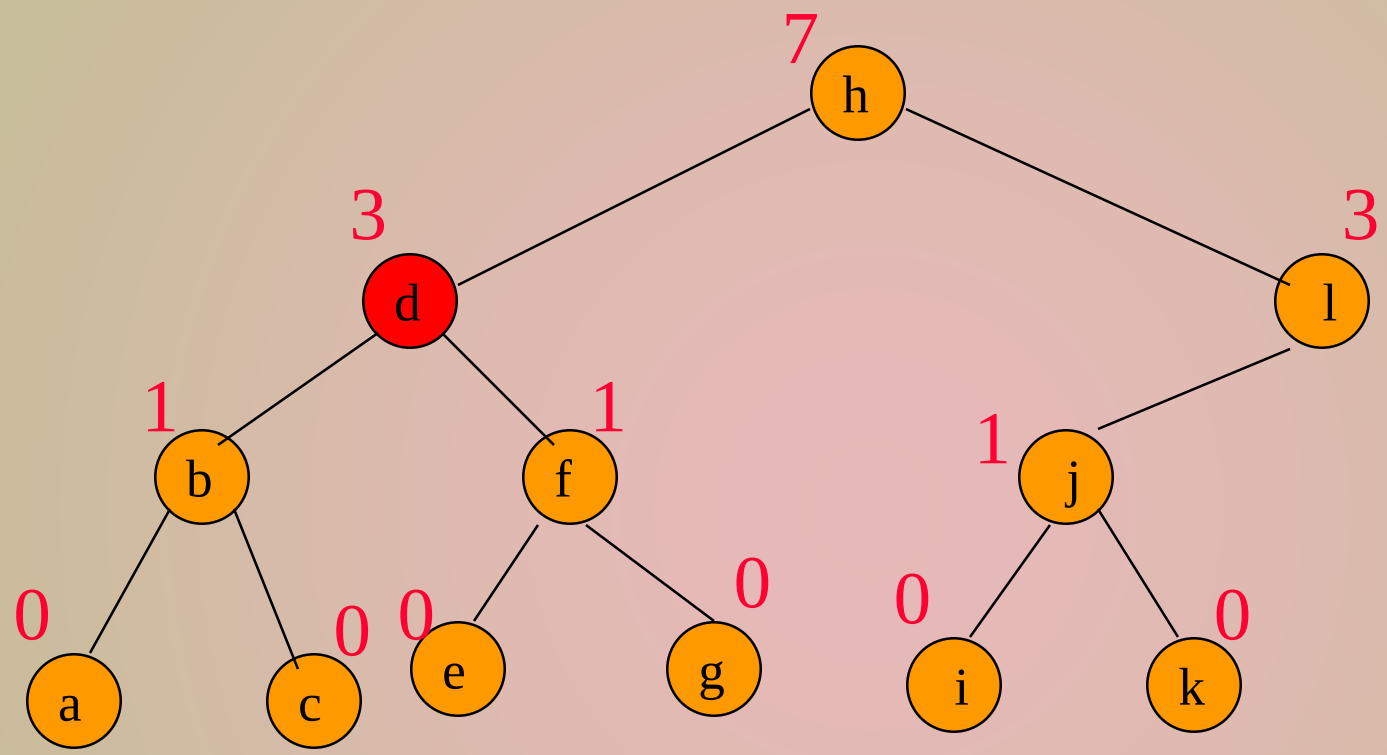

N
index
index_1
index+1

مثال اول: Delete(3)



list = [a,b,c,d,e,f,g,h,i,j,k,l]

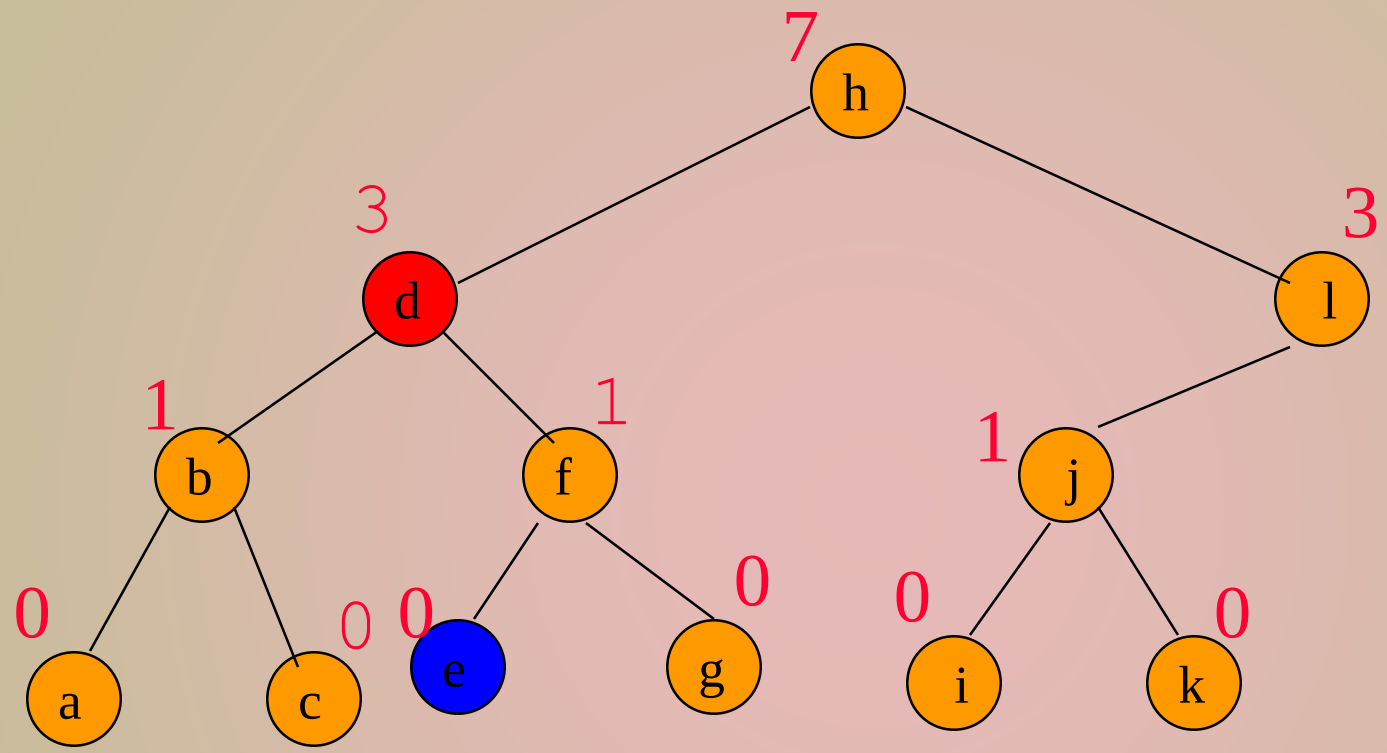
Delete(3)



list = [a,b,c,d,e,f,g,h,i,j,k,l]

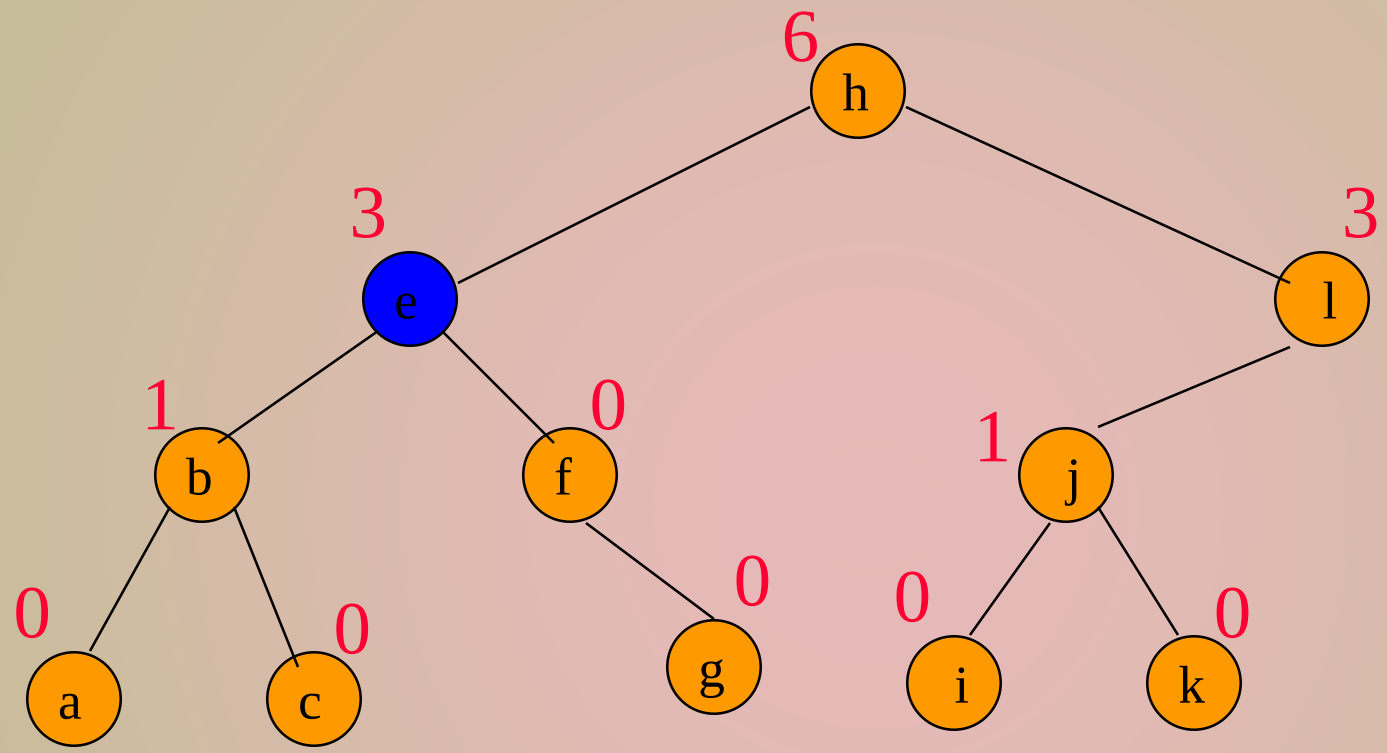
find node with element 3 (d)

راه حل اول Delete(3)



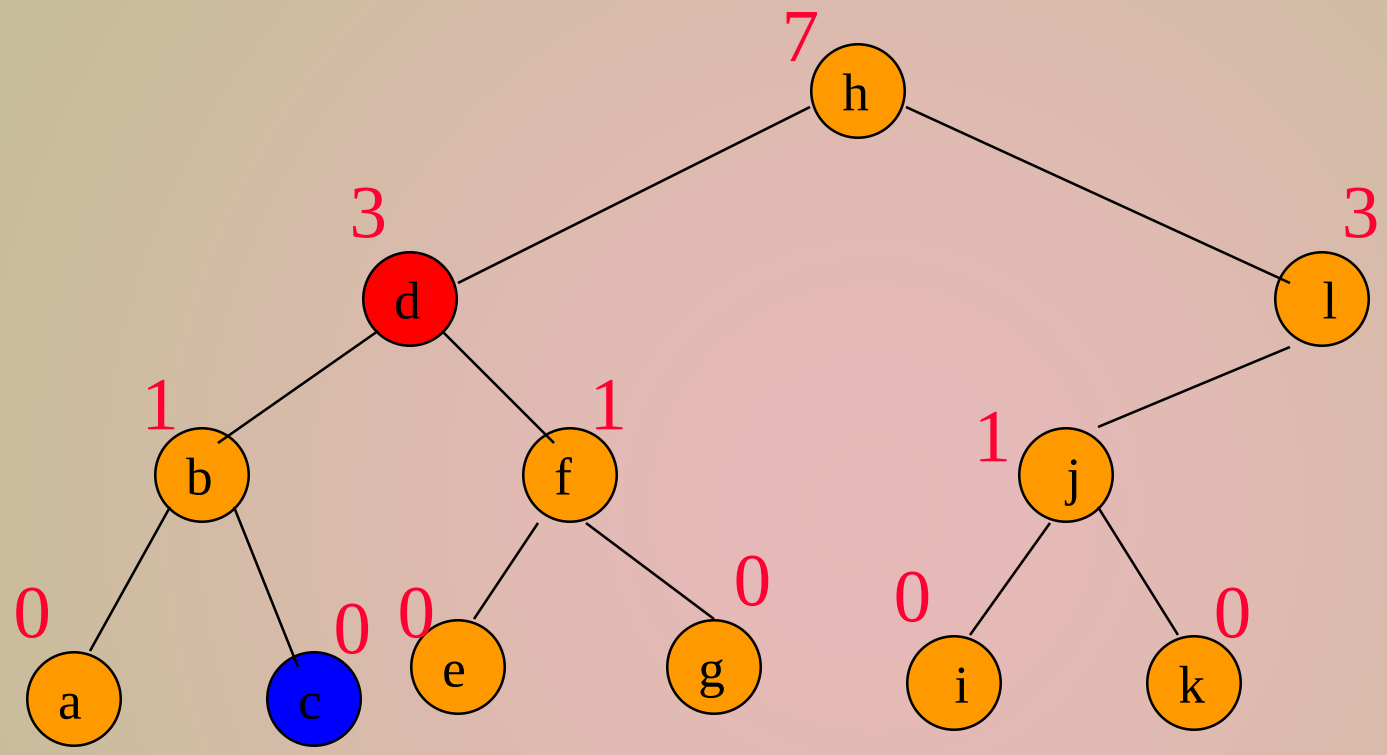
اولین گره مشاهده شده در زیردرخت راست d را در پیمایش میان ترتیب (یعنی e) جایگزین d می کنیم.

راه حل اول Delete(3)



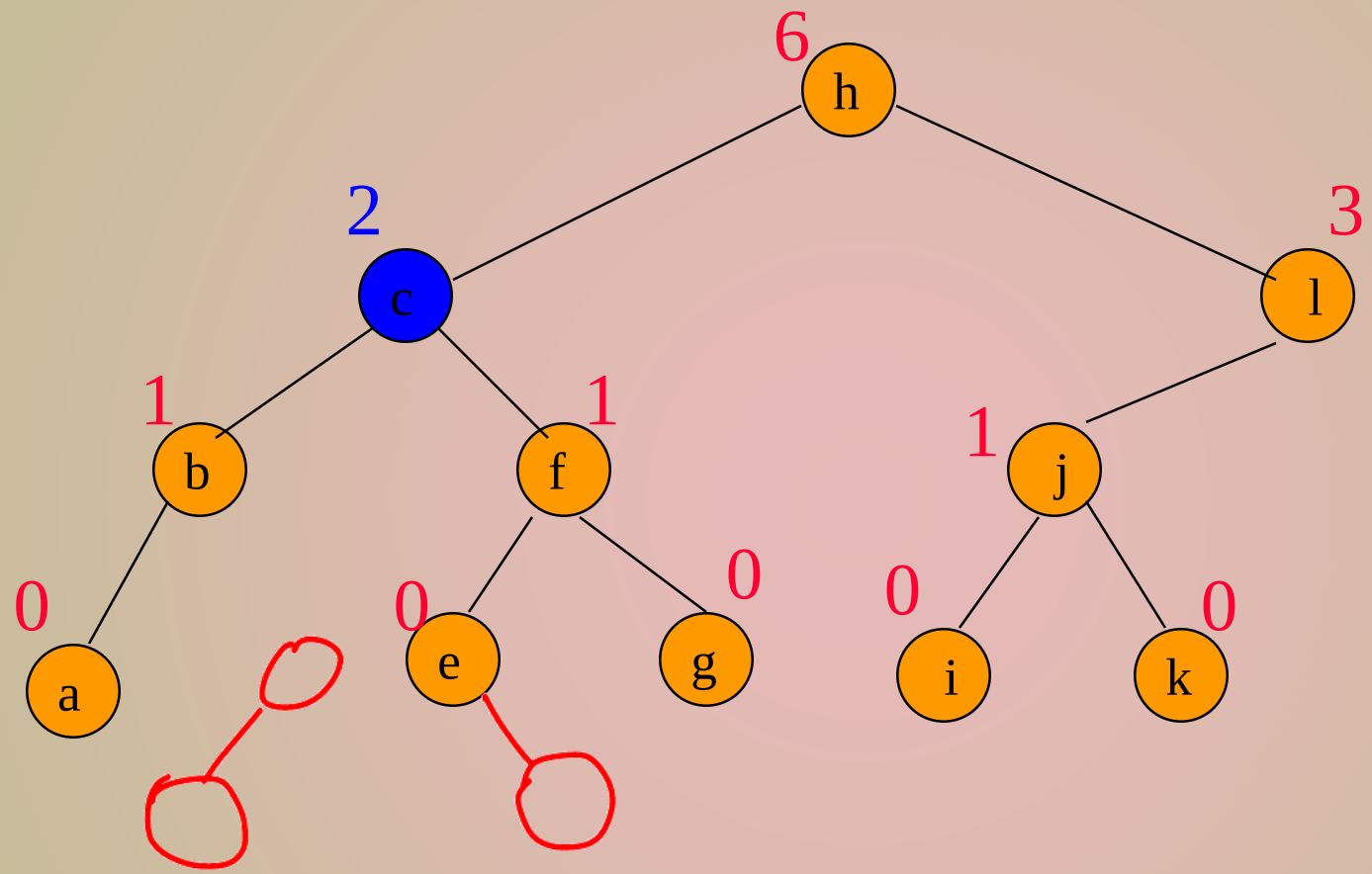
اولین گره مشاهده شده در زیردرخت راست e را در پیمایش میان ترتیب جایگزین d می کنیم.

راه حل دوم Delete(3)



آخرین گره مشاهده شده در زیردرخت چپ d را در پیمایش میان ترتیب (یعنی c) جایگزین d می کنیم.

راه حل دوم Delete(3)



آخرین گره مشاهده در زیردرخت چپ را جایگزین d می کنیم.