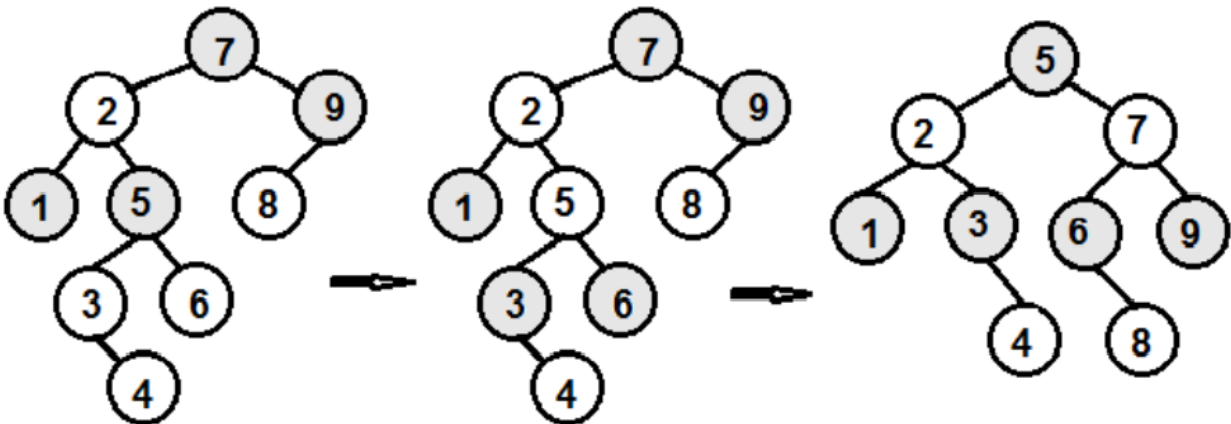


9.2 نمونه سوالات مربوط به بخش

9.2.1 عناصر 4,6,3,1,8,1,2,9,7 را به ترتیب در یک درخت قرمز-سیاه تهی درج کنید. (با ذکر مراحل)

راه حل :

3 مرحله ی پایانی :



9.2.2 نشان دهید در هر درخت قرمز-سیاه ارتفاع هر راس حداکثر 3 برابر کوتاهترین فاصله از این راس به یکی از برگها است.

راه حل :

کوتاهترین فاصله x تا برگ : $d(x)$ ، ارتفاع : $h(x)$

$Bh(x) \geq h(x)/2$ در غیر این صورت حداقل 2 راس قرمز مجاور میشوند . می دانیم $d(x) \geq bh(x)$ بنابراین $2 \cdot d \geq h$

9.2.3 دو درخت قرمز-سیاه با اندازه های n و m داریم، چگونه این دو درخت را در زمان بهینه با هم ادغام

کنیم؟

راه حل :

زمان $O(m+n)$

اگر درختها را به صورت inorder پیمایش کنیم میتوانیم عناصر هر کدام از آنها را به صورت مرتب شده در 2 لیست در $O(m)+O(n)$ به دست آوریم سپس دو لیست به دست آمده را با هم merge میکنیم به کمک آنها در $O(m+n)$ یک درخت جدید بسازیم. روش ساختن درخت جدید :

عنصر میانه لیست را به دست میآوریم و به عنوان ریشه درخت جدید قرار میدهیم و زیردرخت چپ و راست را به همین شکل به صورت

بازگشتی به دست میآوریم : $T(n+m) = O(1) + 2T(n+m/2) \rightarrow \text{overall} : O(n+m)$

9.2.4 الگوریتمی بنویسید که یک BST را رنگ کند تا قرمز-سیاه شود یا تشخیص دهد این کار ممکن نیست.

راه حل :

```
3. For all vertexes like n starting by the root:
5.   if n is root,
6.       color(n) = black
7.       bh(n) = ⌈ h(n)/2 ⌋
8.   else if p(n) is red,
9.       color(n) = black
10.      bh(n) = bh(p(n)).
11.
12.  else (p(n) is black)
13.      if d(n) < bh(p(n)), then
14.          error "shortest path was too short"
15.      else if d(n) = bh(p(n)) then
16.          Color(n) = black
17.      else (d(n) > bh(p(n))
18.          Color(n) = red
19.      either way,
20.          bh(n) = bh(p(n)) - 1
```

$bh(n)$ - سیاه ارتفاع n

$h(n)$ - ارتفاع n

$p(n)$ - والد n

رنگ ریشه را برابر سیاه قرار می‌دهیم و bh آن را برابر با نصف ارتفاع آن قرار می‌دهیم (یعنی bh را برابر حداقل مقدار ممکن قرار می‌دهیم). به ازای هر راس n اگر پدر n قرمز باشد رنگ n سیاه می‌شود $bh(n)$ برابر با bh پدرش می‌شود. اگر پدر n سیاه باشد:

$bh(n)$ برابر با $bh(p(n)) - 1$ می‌شود. اگر ارتفاع n کمتر از سیاه‌ارتفاع پدرش باشد، نمیتوان درخت را رنگ‌آمیزی کرد. اگر ارتفاع n دقیقاً برابر سیاه‌ارتفاع پدرش باشد، n را سیاه می‌کنیم (n و تمام رئوسی که در مسیر n تا برگ قرار دارند باید سیاه شوند). اگر ارتفاع n بیشتر از سیاه‌ارتفاع پدرش باشد، رنگ آن را قرمز می‌کنیم.

(برای یک درخت ممکن است چندین رنگ‌آمیزی مختلف وجود داشته باشد)

9.2.5 آیا امکان دارد که تمام گره‌های یک درخت قرمز - سیاه به رنگ سیاه باشد؟

راه حل :

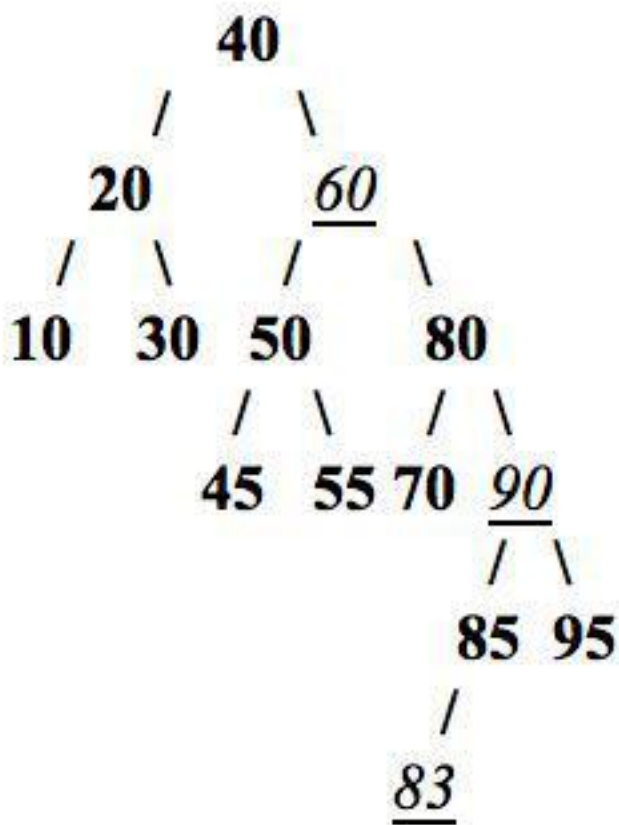
9.2.6 درخت قرمز - سیاه ای مثال بزنید که AVL نباشد!

راه حل :

9.2.7 حداکثر ارتفاع درخت قرمز - سیاه با ۱۴ گره جقدر است؟ مثالی از درختی قرمز - سیاه با ۱۴ گره بکشید که به حداکثر ارتفاع رسیده باشد.

راه حل :

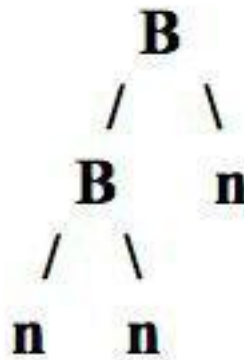
ماکسیمم ارتفاع ۵ است | توجه به راهنمایی سوال یک درخت با black-height ۲ میتواند دقیقا ارتفاع ۵ داشته باشد اگر که به صورت تناوبی آن را تا عمیق ترین گره قرمز و سیاه کنیم.



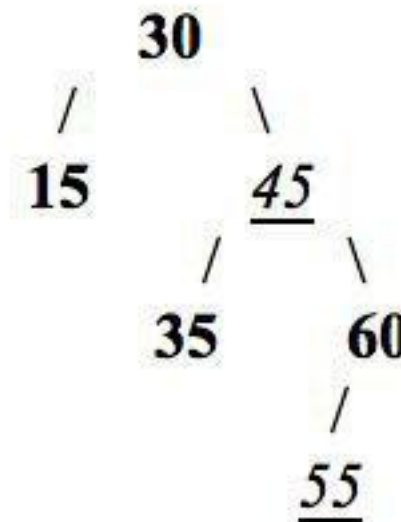
9.2.8 چرا نمیتواند یک درخت قرمز-سیاه ، یک گره مشکمی دقیقا یک فرزند سیاه و هیچ فرزند قرمز داشته باشد؟!

راه حل:

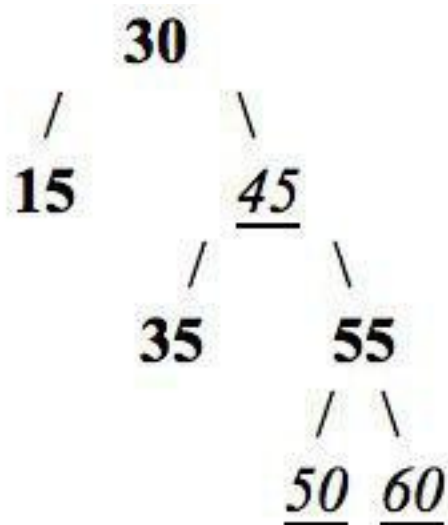
برای اینکه black depth گره های خارجی گره های سیاه نمیتواند برابر black depth بقیه ی گره های خارجی آن گره سیاه باشد با توجه به شکل زیر سیاه ها را با B و گره های خارجی را با n نشان داده شده.



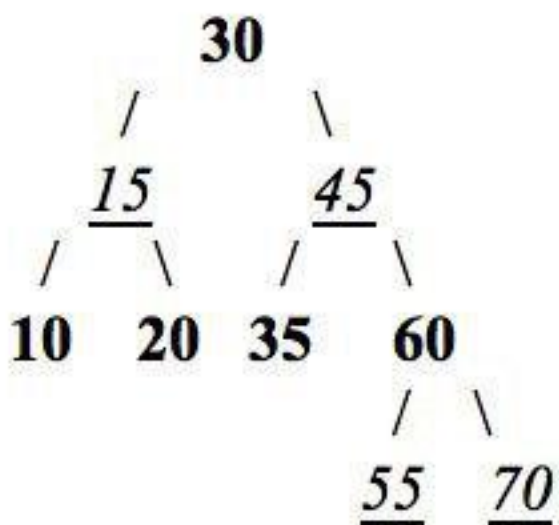
9.2.9 عدد ۵۰ را به درخت قرمز-سیاه زیر اضافه کنید .



راه حل :

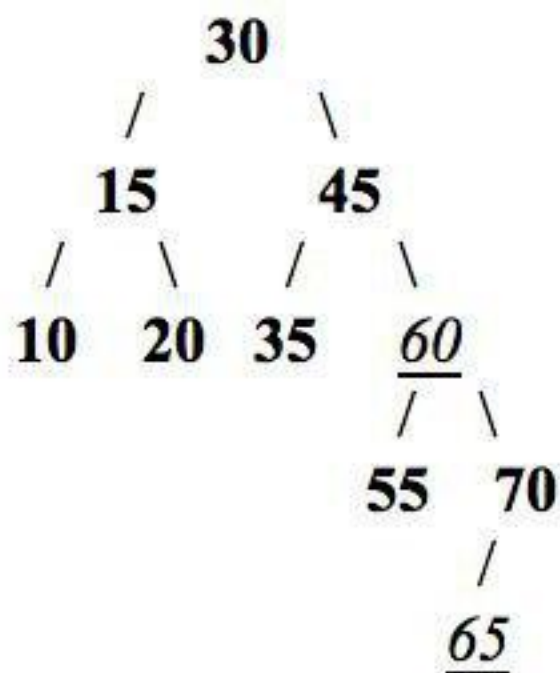


9.2.10 عدد 65 را به درخت قرمز-سیاه زیر اضافه کنید .

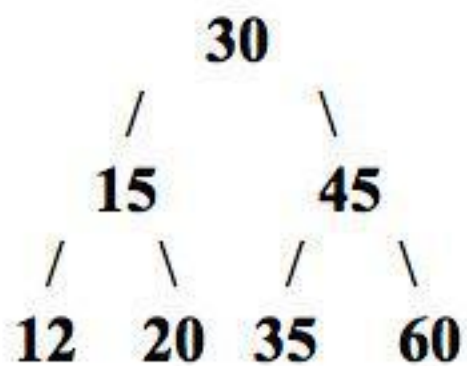


شکل 2

راه حل :

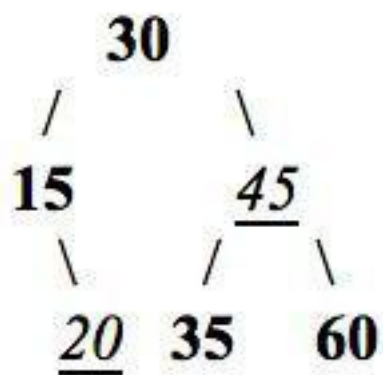


9.2.11 عدد ۱۲ را از درخت قرمز سیاه زیر پاک کنید و نتیجه را نشان دهید.

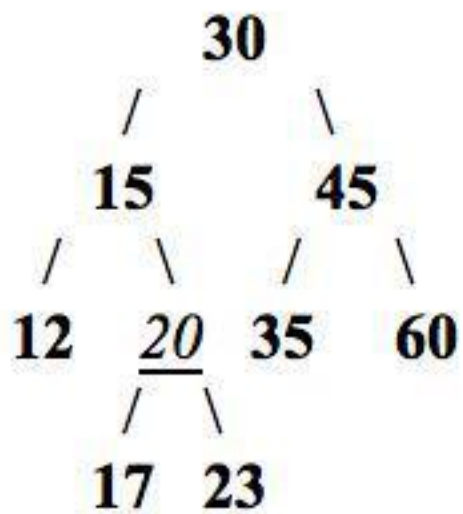


شکل 3

راه حل :



9.2.12 عدد ۱۲ را از درخت قرمز سیاه زیر پاک کنید و نتیجه را نشان دهید.



راه حل :

