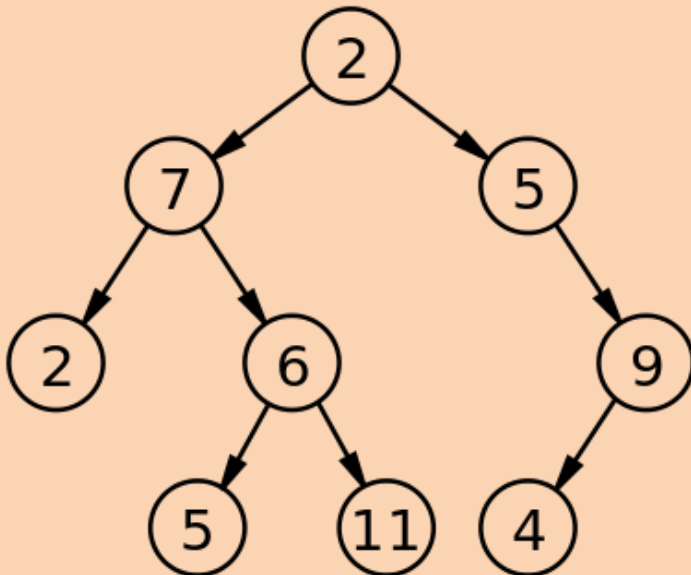




ساختارهای داده

درخت‌های قرمز-سیاه

Red-Black Trees



مدرس:

سید کمال الدین غیاثی شیرازی

B-Trees

- Bayer, R., and E. McCreight. "Organization and Maintenance of Large Ordered Indices." In *Proc. 1970 ACM-SIGFIDENT Workshop Data Description and Access*, pp. 107-141. 1970.
- Bayer, R., and E. M. McCreight. "Organization and maintenance of large ordered indexes." *Acta Informatica* 1, no. 3 (1972): 173-189.

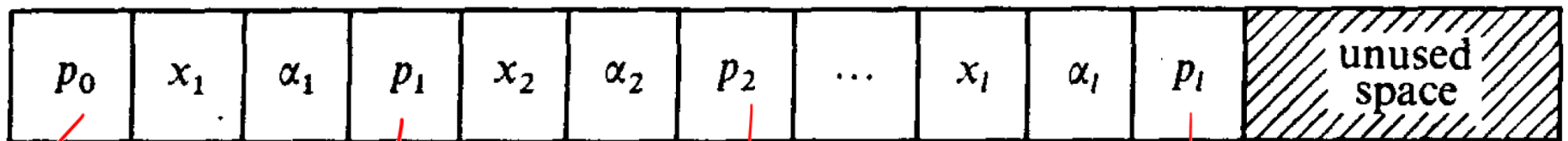


Fig. 1. Organization of a page

B-Trees

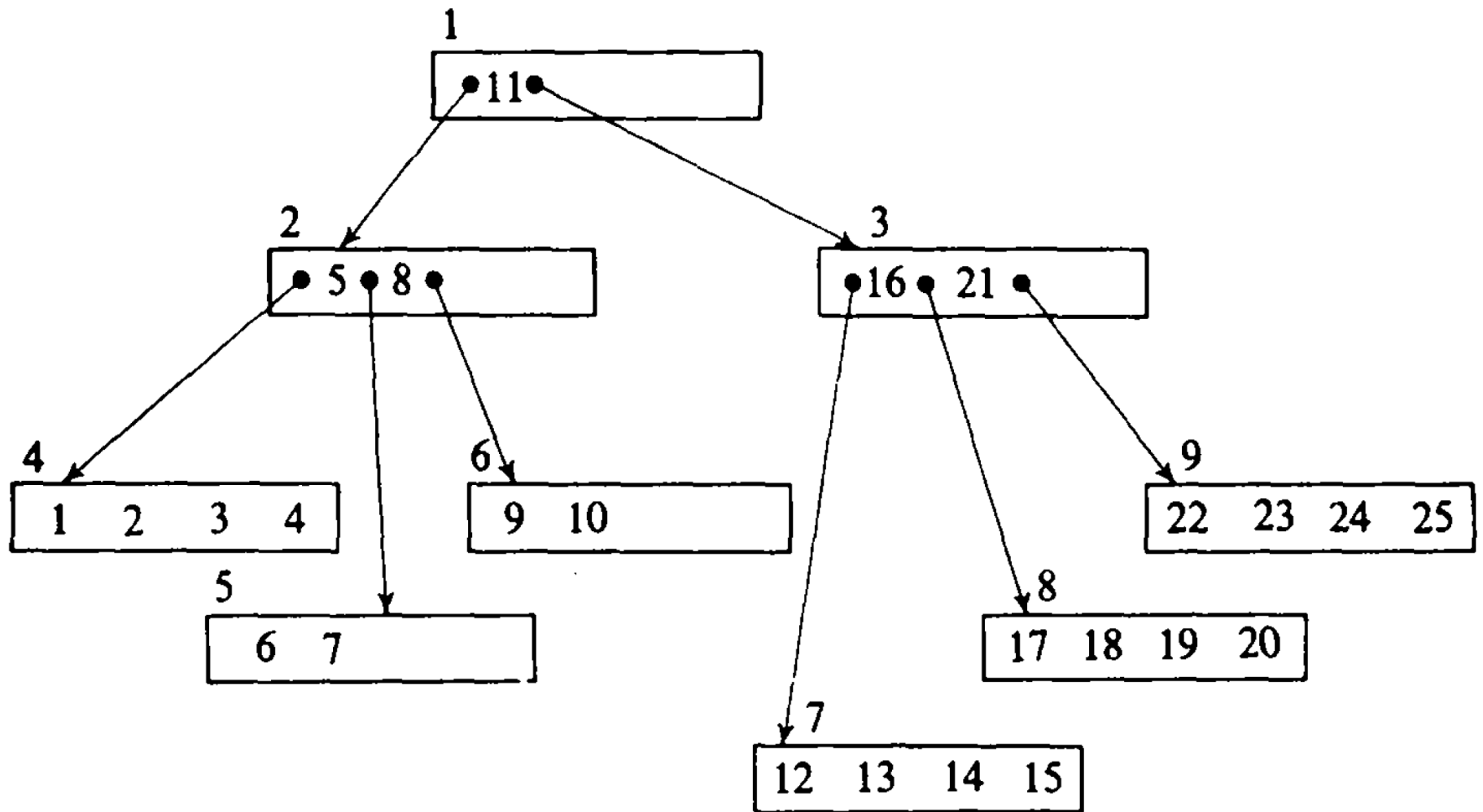
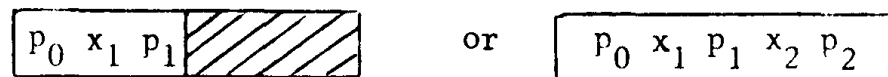


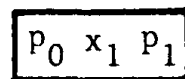
Fig. 2. A data structure in $\tau(2, 3)$ for an index

Binary B-Trees

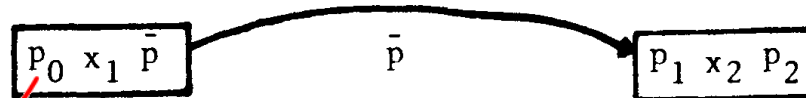
- Bayer, Rudolf. "Binary B-trees for virtual memory." In *Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control*, pp. 219-235. 1971.



where the x_i are keys and the p_i are pointers to other pages in a B-tree. Instead of storing pages as "physical" pages, thus possibly half-empty, we can store them as threaded lists of 1 or 2 elements, each element consisting of one key, two pointers, and, as we shall see, one additional bit. Thus the half-empty page above would be represented as:



and the full page would be represented as:



Clearly 1 bit is needed to indicate, whether the right pointer in such an element is a "horizontal" (ρ -pointer) pointer to the second element of a "full" page or a "vertical" pointer (δ -pointer) to a page in the next level of the B-tree.

Symmetric Binary B-Trees

- Bayer, Rudolf. "Symmetric binary B-trees: Data structure and maintenance algorithms." *Acta informatica* 1, no. 4 (1972): 290-306.

- عدم تقارن Binary B-Tree در این بود که فرزند چپ همواره به گرهی در سطح بعد اشاره می‌کند، اما فرزند راست ممکن است ادامه‌ی یک گره B یا گرهی در سطح بعد باشد.

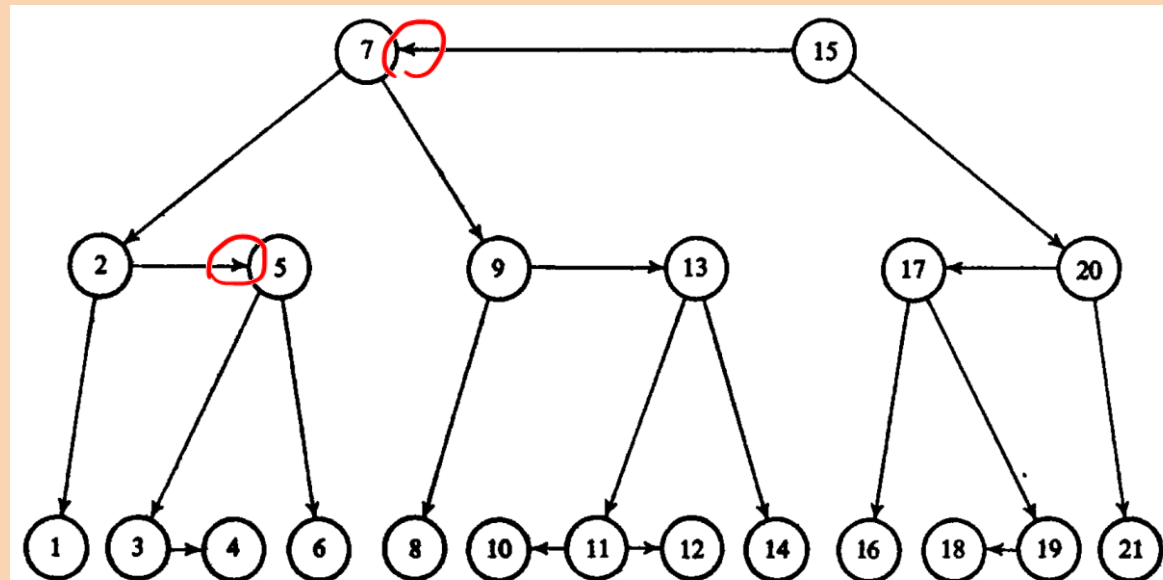
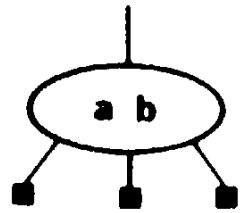


Fig. 1. Example of a symmetric binary B-tree

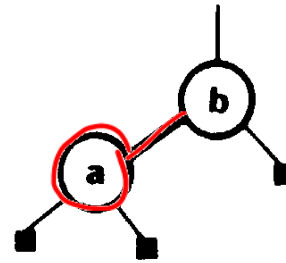
Red-Black Trees

- Guibas, Leo J., and Robert Sedgwick. "A dichromatic framework for balanced trees." In *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, pp. 8-21. IEEE, 1978.

- در این مقاله پیشنهاد استفاده از دو رنگ قرمز و سیاه برای نمایش گره‌های درخت متوازن مطرح می‌شود. می‌توان گفت این مرجع اصلی درخت‌های قرمز-سیاه است.
- درخت‌های 2-3 و 2-3-4 در نظر گرفته می‌شوند و درخت‌های قرمز-سیاه معادل آنها معرفی می‌شود.
- درخت‌های قرمز-سیاه 2-3 بخاطر تعداد چرخش زیادتر در درج کنار گذاشته می‌شوند.
- درخت‌های قرمز-سیاه 2-3-4 به عنوان درخت‌های قرمز-سیاه شناخته می‌شوند.



corresponds to



or

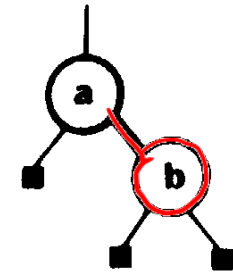
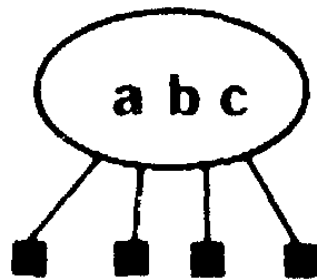
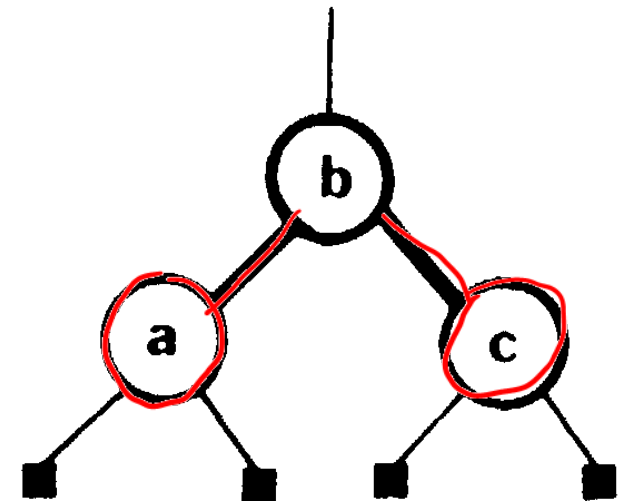


Figure 1. The binarization of a 3-node



corresponds to



درخت‌های قرمز-سیاه در درس استاد Sartaj Sahni

- Sahni, Sartaj. "Data structures." *Algorithms, and Applications in C++, WCB McGraw-Hill* (1998).
- Sahni, Sartaj. *Data structures, algorithms, and applications in Java*. Universities Press, 2005.
- الگوریتم اولیه‌ی درج و حذف در درخت‌های قرمز-سیاه بسیار پیچیده است و حالات زیادی را در نظر می‌گیرد که هر کدام پیچیدگی بسیار زیادی دارند (بویژه در مورد حذف).
- این الگوریتم را در فیلم جداگانه‌ای شرح می‌دهم (مشاهده اختیاری).
- <https://www.cise.ufl.edu/academics/courses/preview/cop5536sahni>
- <https://www.cise.ufl.edu/~sahni/cop5536/index.html>
- <https://www.cise.ufl.edu/~sahni/cop5536/presentations.htm>

درخت‌های قرمز-سیاه در کتاب CLRS

- Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

- در این کتاب بدون ارجاع به هیچ مقاله‌ای، یک الگوریتم ساده‌تر برای انجام عملیات درج و حذف بر روی درخت‌های قرمز-سیاه ارائه می‌شود.
- این الگوریتم را در فیلم جداگانه‌ای شرح می‌دهم.

Left-Leaning Red-Black Trees

- Sedgewick, Robert. "Left-leaning red-black trees." In *Dagstuhl Workshop on Data Structures*, vol. 17. 2008. URL: <http://www.cs.princeton.edu/~rs/talks/LLRB/LLRB.pdf>

● این الگوریتم را در فیلم جداگانه‌ای شرح می‌دهم.

Parity-Seeking Deletion Algorithm

- Ghiasi-Shirazi, Kamaledin, Taraneh Ghandi, Ali Taghizadeh, and Ali Rahimi-Baigi. "A Pedagogically Sound yet Efficient Deletion algorithm for Red-Black Trees: The Parity-Seeking Delete Algorithm." (2020).

● ما در این مقاله نشان می‌دهیم که

○ درخت‌های قرمز-سیاه ۲-۳، که در ابتدا بخاطر تعداد چرخش بالاتر کنار گذاشته شده بودند،

بسیار کارآمد هستند.

○ یک الگوریتم جدید حذف بسیار ساده، قابل فهم، و در عین حال کارا ارائه می‌کنیم.

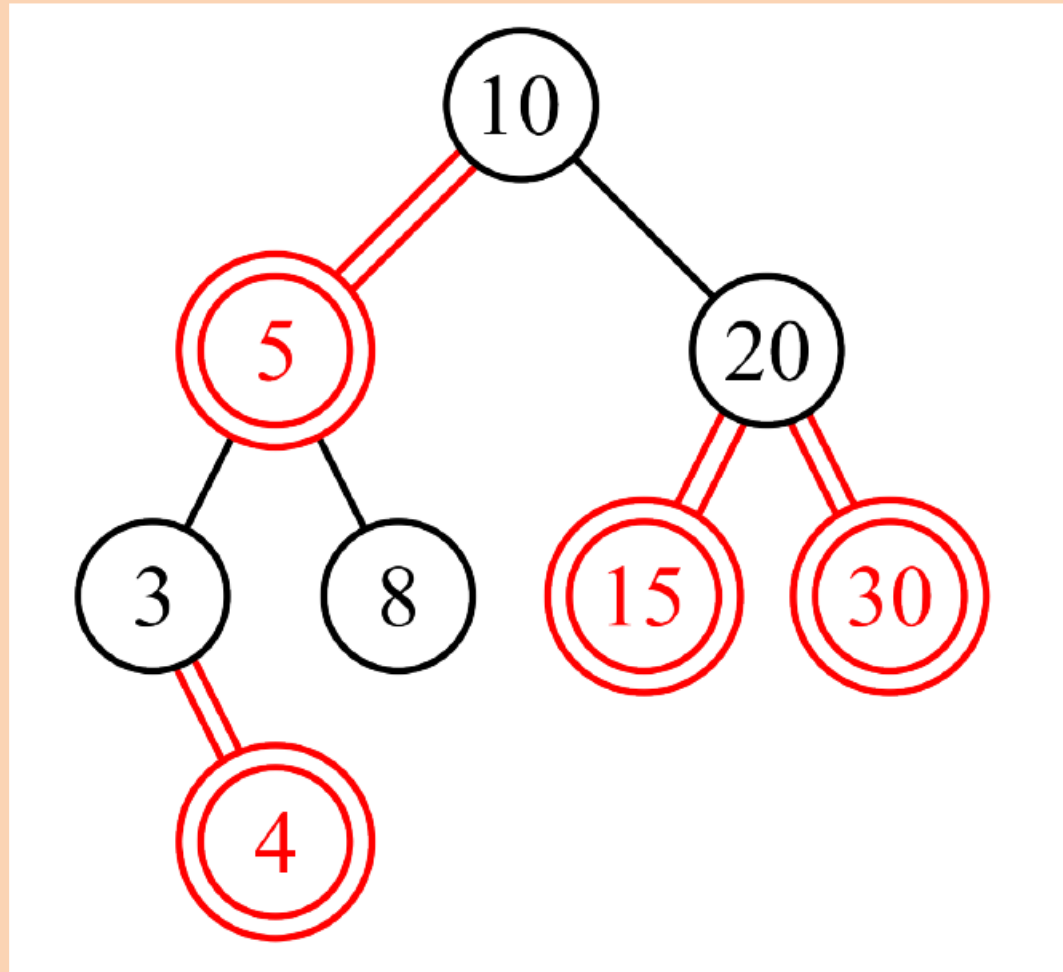
○ الگوریتم حذف ما برای درخت‌های قرمز-سیاه ۲-۳-۴ با الگوریتم حذف CLRS معادل است.

● این الگوریتم را در فیلم جداگانه‌ای شرح می‌دهم.

تعریف درخت‌های قرمز-سیاه

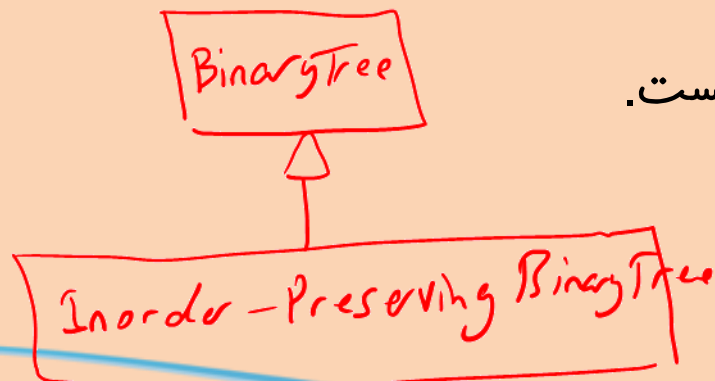
- یک درخت قرمز-سیاه یک درخت جستجوی دودویی است که گره‌ها دارای یک ویژگی اضافه به نام رنگ نیز می‌باشند. رنگ یک گره می‌تواند **قرمز** یا سیاه باشد.
- درخت‌های قرمز-سیاه با ویژگی‌های زیر شناخته می‌شوند:
 - ریشه سیاه است.
 - اگر گرهی قرمز باشد، والد آن سیاه است.
 - تمام مسیرها از ریشه تا برگ، دارای تعداد یکسانی گره سیاه است.
- به تعداد گره‌های سیاه از ریشه تا برگ، ارتفاع سیاه (black-height) درخت گفته می‌شود.

مثالی از يك درخت قرمز-سیاه

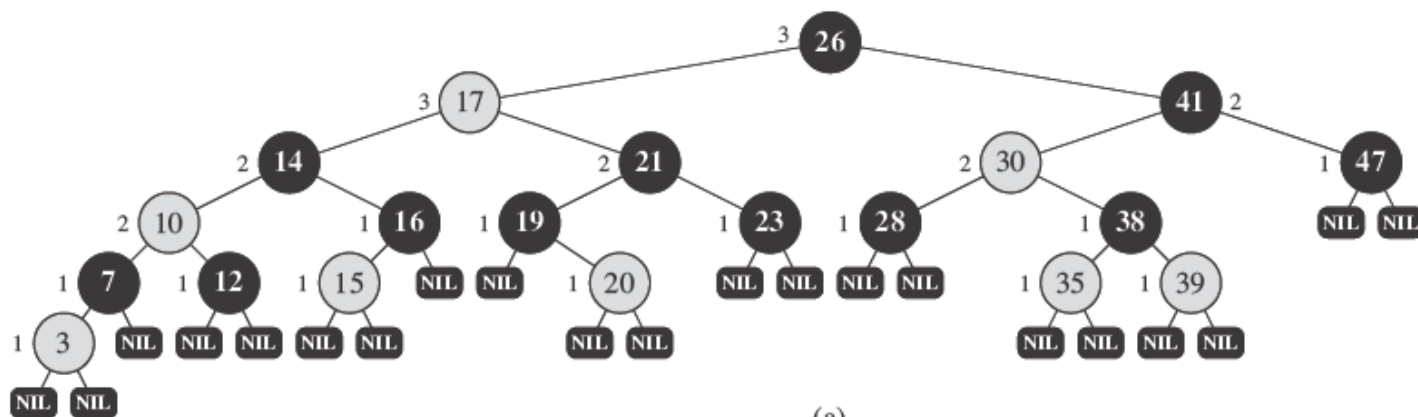


توجه مهم

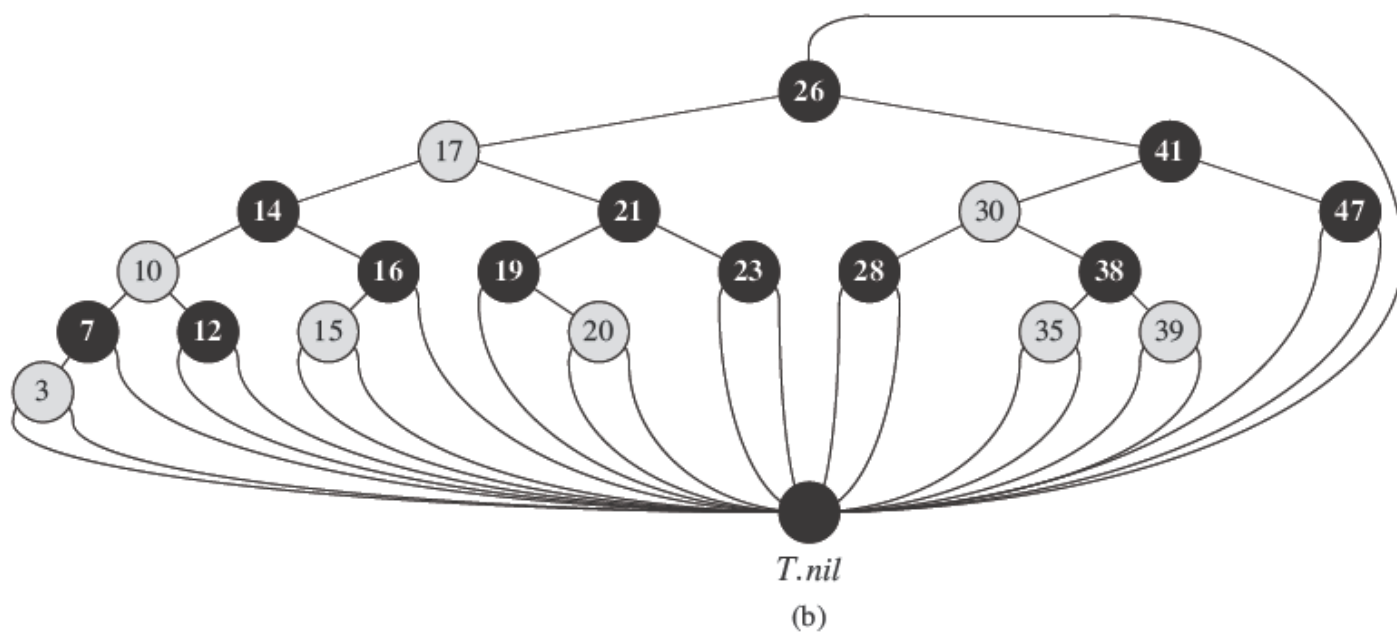
- این تعریف که از کتاب CLRS گرفته شده است، درخت‌های قرمز-سیاه را محدود به درخت‌های جستجوی دودویی می‌کند.
- اما این صحیح نیست و در حالت کلی ویژگی اصلی درخت‌های قرمز-سیاه این است که با وجود تغییر درخت با هدف حفظ توازن، پیمایش میان‌ترتیب را حفظ می‌کنند.
- استاد Sartaj Sahni در درس و کتاب خود تعریف دقیق‌تری برای درخت‌های قرمز-سیاه ارائه می‌کنند که محدود به جستجوی دودویی نیست.



نمایش صریح گره‌های خارجی با یک گره مشترک سیاه به نام nil



(a)



(b)

درخت‌های قرمز-سیاه با درخت‌های 2-3-4 معادل هستند

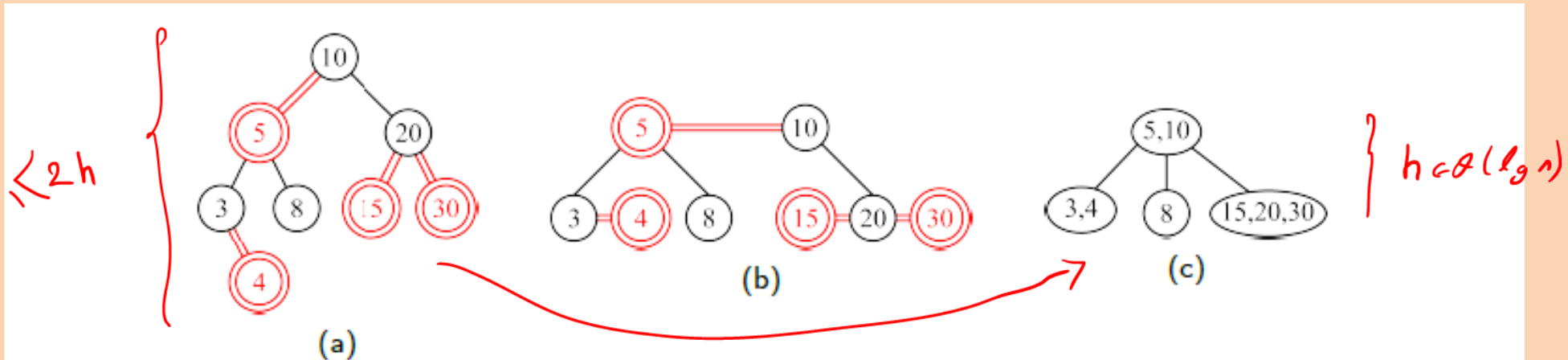
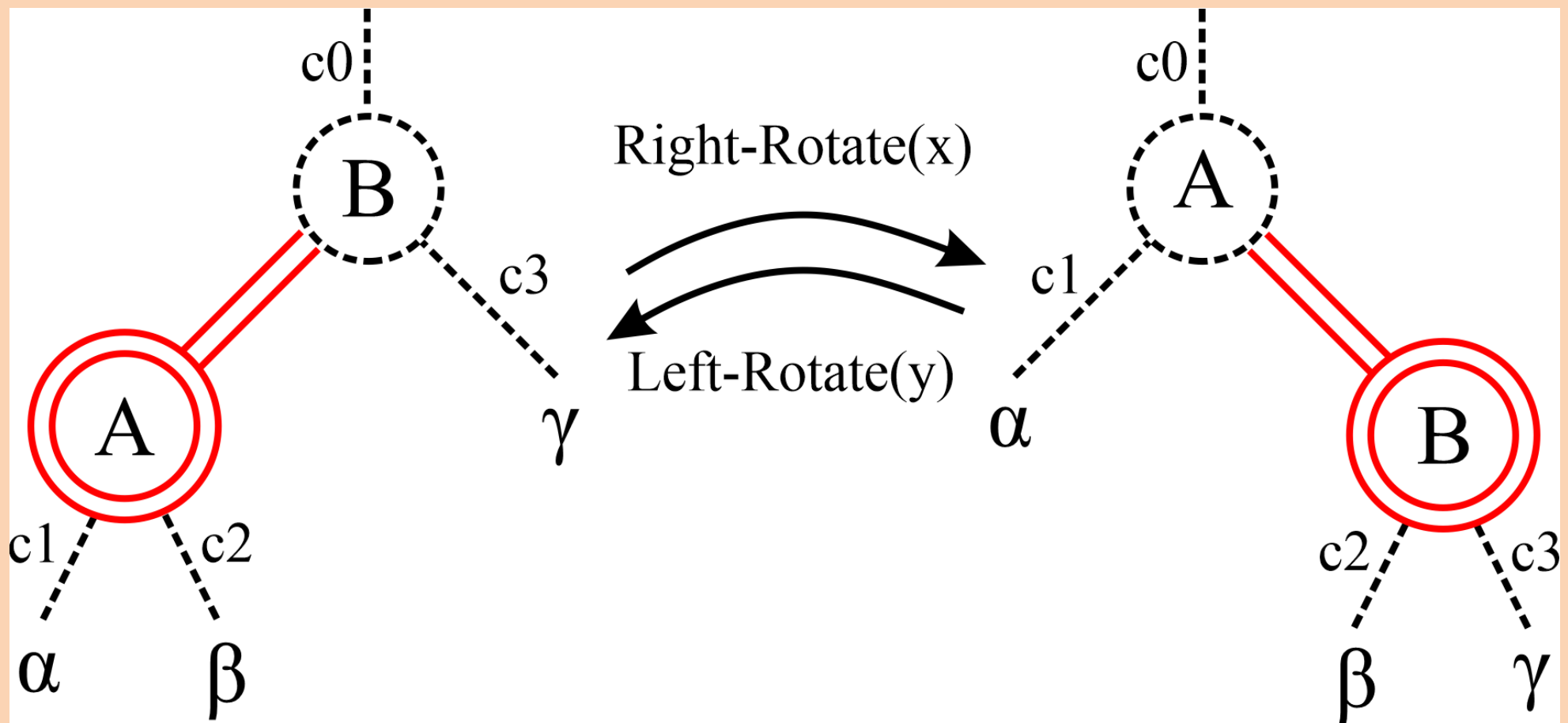


Figure 1: (a) An RB tree, (b) its representation with horizontal red links, and (c) its equivalent 2-3-4 tree.

- ارتفاع درخت قرمز-سیاه حداکثر دو برابر ارتفاع درخت 2-3-4 معادل است.
- بنابراین ارتفاع درخت قرمز-سیاه از مرتبه‌ی $\lg n$ است.
- می‌توان نشان داد که ارتفاع درخت قرمز-سیاه حداکثر $\lfloor 2 \lg(n + 1) \rfloor$ است.

عملیات چرخش به چپ و چرخش به راست



$\alpha, A, \beta, B, \gamma$

$\alpha, A, \beta, B, \gamma$

عمل برگرداندن رنگ (Color Flip)

