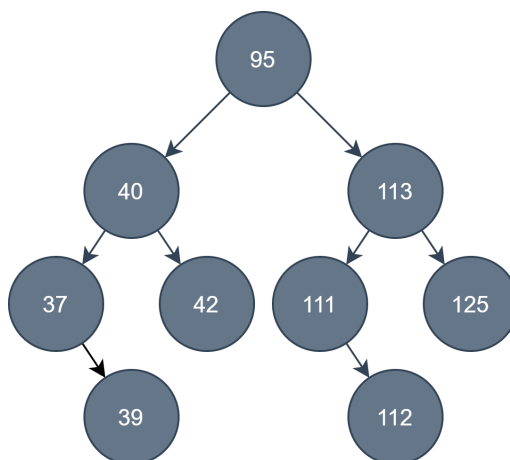


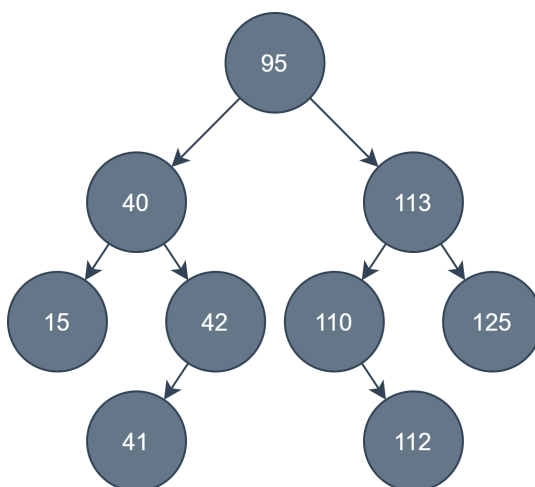
بسمه تعالی

- تمرین سری سوم درس ساختمان داده ها و مبانی الگوریتم ها
 - پاسخ تمرین در قالب یک فایل pdf تایپ شده یا دست نویس اسکن شده (مرتب و خوانا) و با نام StudentNumber_HW3.pdf آپلود شود.
 - مهلت ارسال تمرین تا ساعت ۱۱:۵۹ روز جمعه مورخ ۷ آذر ۱۳۹۹ می باشد.
 - در صورتی که درمورد این تمرین سوال یا ابهامی داشتید با ایمیل dsfall1399@gmail.com با تدریس یاران در ارتباط باشید. لطفا برای ایمیل زدن فرمت زیر را در قسمت subject رعایت کنید:
برای سوال از مباحث مختلف:
«سوال_اسم مبحث» (مثال: «سوال_درخت جستجو»)
برای سوال از یک تمرین خاص:
«تمرین_شماره تمرین_شماره سوال» (مثال: «تمرین_۳_۳»)
- همچنین خواهشمند است در متن ایمیل به شماره دانشجویی خود اشاره کنید.

۱- درخت باینری زیر را در نظر بگیرید، بعضی از مقادیر هستند که اگر به درخت اضافه شوند، ارتفاع درخت را زیاد می‌کنند. تمام مقداری صحیحی که این موضوع برای آنها صدق می‌کند را پیدا کنید. اگر چنین مقداری وجود ندارد، علت را توضیح دهید. توجه کنید که باید هر مقدار به تنهایی ارتفاع درخت را زیاد کند و نباید چندین بار به درخت مقداری اضافه کرد.



۲- برای یک مجموعه از مقادیر، درخت‌های باینری زیادی وجود دارد. اگر اعمال rotation مانند آنچه که برای درخت red/black تعریف می‌شود را روی هر عنصر دلخواه یک درخت باینری اعمال کنیم به درخت جدیدی تبدیل می‌شود. برای درخت زیر، درخت باینری دیگری با همین مقادیر پیشنهاد دهید که با اعمال دنباله‌ای از چرخش‌ها (left/right rotation) به آن تبدیل نشود. اگر چنین درختی وجود ندارد علت را توضیح دهید.



۳- یک الگوریتم غیربازگشتی برای پیمایش inorder و postorder و preorder درخت باینری ارائه دهید. (شبه کد و تحلیل زمانی الزامیست)

۴- فرض کنید دو درخت دودویی جستجوی متوازن T_1 و T_2 به ترتیب دارای n_1 و n_2 رأس، مجموعه‌های S_1 و S_2 را نمایش می‌دهند

الف) الگوریتمی ارائه دهید که در مرتبه ی زمانی $O(n_1 \log(n_2))$ و با استفاده از حافظه‌ی اضافی $O(1)$ مشخص کند که آیا $S_1 \subseteq S_2$ هست یا خیر.

ب) الگوریتمی ارائه دهید که در مرتبه ی زمانی $O(n_1 + n_2)$ و با استفاده از حافظه‌ی اضافی $O(n_1 + n_2)$ مشخص کند که آیا $S_1 \subseteq S_2$ هست یا خیر.

۵- الگوریتمی ارائه دهید که در مرتبه زمانی $O(n)$ بررسی کند که آیا یک درخت دودویی، درخت دودویی جستجو هست یا خیر.

۶- دو دنباله از پیمایش inorder و postorder یک درخت دودویی داریم. درخت اصلی را رسم کنید.

Inorder traversal = D,H,B,E,A,F,C,G

Postorder traversal = H,D,E,B,F,G,C,A

راهنمایی ۱: در پیمایش postorder ریشه در کجای دنباله قرار دارد؟

راهنمایی ۲: اگر ریشه را در دنباله‌ی بدست آمده از inorder پیدا کردیم، عناصر زیردرخت سمت چپ و راست ریشه، در

آرایه، چه وضعیتی نسبت به ریشه دارند؟

۷- با ورود مقادیر زیر به ترتیب یک MinHeap بسازید.

2,50,45,70,75,22,40,55,45

توجه کنید که اعداد به ترتیب وارد می‌شوند و فقط یک درخت MinHeap است که از این ترتیب به دست می‌آید.

۸- مقادیر زیر را به ترتیب وارد یک درخت red/black بکنید. مراحل و case ها را در هر مرحله توضیح دهید. (اولین بار

عدد ۷ وارد می‌شوند)

7,3,18,10,22,8,11,26,2,6,13

سپس مقادیر زیر را به ترتیب حذف کنید. مراحل و case ها را در هر مرحله توضیح دهید. (اولین بار عدد ۱۸ پاک می‌شود)

18,11,3,10,22

امتیازی:

۹- ثابت کنید ارتفاع هر درخت red/black با n عنصر از $2 \times \log_2(n+1)$ کمتر است. توضیح دهید چرا این ویژگی

دلیل برتری درخت red/black بر درخت باینری است.