



دانشگاه مهندسی کامپیوتر

ساختمان داده

نیمسال اول ۹۸-۹۹

امتحان عملی شماره ۲-الف

محمدجواد پیرهادی

مطهره میرزائی

سید صالح اعتمادی

دوشنبه ۲ دی ماه

نامگذاری ها

این تمرین باید

- روی شاخه fb_Exam2a پیاده سازی شود.
- در پوشه E2a در ریشه ریپازیتوری گیت پروژه DS98991
- نام E2a:solution
- نام پروژه اصلی: E2a
- نام پروژه تست: E2a.Tests
- تمرین باید در این شاخه add/commit/push بشود و پول ریکوست برای بردن به مستر درست شود و بعد از بیلد موفقیت آمیز "کامل" شود. برای اطمینان از کامل شدن لازم است روی کامپیوتر خودتون روی شاخه مستر "پول" انجام دهید و از تکمیل پول ریکوست اطمینان پیدا کنید.
- تمام نام های کلاس ها، متدها، شاخه ها... باید عینا با رعایت حروف بزرگ و کوچک انجام شود.
- در این بخش از امتحان استفاده از کد تمرین هایی که انجام داده اید و هر کد آماده ی دیگر مجاز نیست.

ساخت پول ریکوئست - ۱۰٪

بعد از درست کردن پروژه ها و اضافه کردن فایلها قبل از اینکه هیچ کدی بنویسید لازم است که فایل ها را به گیت add/commit/push کنید و در visualstudio.com برای بردن این شاخه به مستر یک پول ریکوست به نام Exam2a درست کنید. این قسمت به تنهایی ۱۰ درصد نمره را دارد.

سوال ۱ - پیمایش InOrder در درجستجوی دودویی

می خواهیم یک BST یا Binary Search Tree را به صورت InOrder پیمایش کنیم. در این صورت نتیجه Sort شده ی اعداد موجود در درخت است. این سوال به منظور تسهیل دیباگ کردن و چک کردن جوابها به این شکل طراحی شده است. **ولی sort کردن اعداد به هیچ وجه قابل قبول نمی باشد و در صورت استفاده، کل نمره امتحان از بین خواهد رفت.**

ورودی و خروجی:

ورودی طول آرایه و یک درخت BST است که به صورت آرایه به شما داده شده است. BST داده شده بالانس می باشد اما لزوما Complete نیست. و -1 های موجود نشان دهنده ی این است که آن نود وجود ندارد. خروجی پیمایش InOrder این درخت می باشد.

نمونه ورودی:	نمونه خروجی:
15 8 3 10 -1 5 -1 14 -1 -1 -1 -1 -1 -1 -1	3 5 8 10 14
15 6 1 10 -1 4 -1 14 -1 -1 -1 -1 -1 -1 -1	1 4 6 10 14

سوال ۲ – دوتا بچه کافی نیست 😊: درخت MinHeap سه فرزنده

یک MinHeap به شما داده می شود که هر نود آن سه فرزند دارد و از شما خواسته می شود که مقدار نود گفته شده را با عدد مشخصی جمع کرده و مقدار جدید را جایگزین مقدار قبلی کنید. طبیعی است که بعد از این تغییر ممکن است قواعد درخت Heap در این محل نقض شده باشند. لازم است با جابجایی مناسب مقادیر در درخت درخت را مجدد به حالت MinHeap آوریم.

ورودی و خروجی:

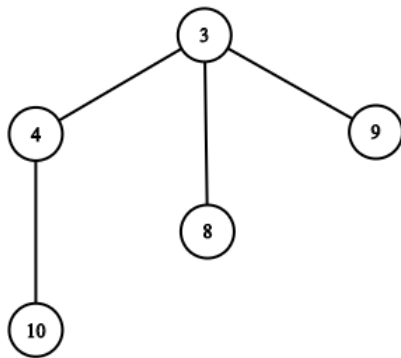
ورودی یک Min-Heap است که خط اول تعداد نود هاست و اولین عدد در خط دوم index نودی است که قرار است تغییر کند و عدد دوم مقداری است که باید با نود جمع کنید و خط سوم هم MinHeap است که به صورت آرایه به شما داده شده است.

خروجی یک آرایه است که شکل جدید Min-Heap را نشان می دهد.

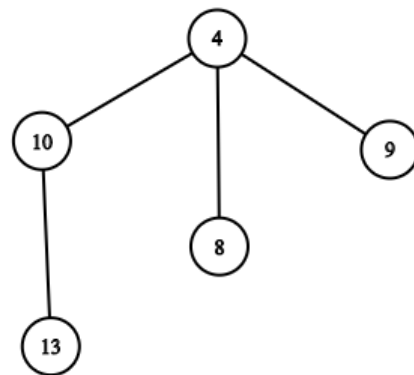
نمونه ورودی:	نمونه خروجی:
5 0 10 3 4 8 9 10	4 10 8 9 13
5 1 11 3 7 8 10 11	3 11 8 10 18

شکل معادل

درخت MinHeap اولیه



درخت MinHeap نهایی



نکات دیگر

- در صورت نیاز میتوانید متدها یا کلاس های جدید تعریف کنید.
- برای اینکه برای فرستادن امتحان در گیت دچار مشکل نشوید قبل از شروع هر کاری:
 - `git checkout master`
 - `git pull`
- مطمئن شوید که شاخه مستر شما هیچ تفاوتی با شاخه مستر سمت سرور ندارد. بعد شاخه جدید را درست کنید:

- `git checkout -b fb_Exam2a`

- بعد از درست کردن پروژه اصلی و تست (مانند تمرین ها) فایل "GradedTests.cs" را به پروژه تست اضافه کرده، بقیه فایل‌های سی شارپ را به پروژه اصلی اضافه کنید. همچنین پوشه TestData را با تمام محتویات آن داخل پروژه تست کپی کرده و مثل تمرین ها تنظیمات زیر را به فایل E2a.Tests.csproj اضافه کنید.

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3 <PropertyGroup>...</PropertyGroup>
8
9 <ItemGroup>...</ItemGroup>
14
15 <ItemGroup>...</ItemGroup>
19
20 <ItemGroup>
21 <Content Include="TestData\**">
22 <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
23 </Content>
24 </ItemGroup>
25
26 </Project>
27
```

