دانشکده مهندسی کامپیوتر

ساختمان‌داده

نیم‌سال اول ۹۸-۹۹

امتحان عملی شماره ۲–ب

محمدجواد پیرهادی

مطهره میرزائی

سید صالح اعتمادی

دوشنبه ۲ دی ماه

نامگذاری ها

این تمرین باید

- روی شاخه fb_Exam2b پیاده سازی شود.
- در پوشه E1b در ریشه ریپازیتوری گیت پروژه DS98991
- نام solution: E2b
- نام پروژه اصلی: E2b
- نام پروژه تست: E2b.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 بزنید 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 Exam2b درست کنید. این قسمت به تنهایی ۱۰ درصد نمره را دارد.

سوال ۱ –پیاده‌سازی Next برای درخت جستجوی دودویی

به عنوان آخرین سوال باید متد Next را برای BST پیاده سازی کنید. یک BST به صورت آرایه به شما داده می شود و شما باید Next نود خواسته شده را پیدا کنید. فرزند چپ عنصری در مکان i در مکان 2i+1 بوده و فرزند راستش در مکان 2i+2. مقدار ۱- برای فرزند هر عنصر به معنای عدم داشتن فرزند می‌باشد.

ورودی و خروجی:

ورودی یک BST به صورت آرایه است که در خط اول تعداد نود ها و index نود خواسته شده و در خط دوم BST به صورت آرایه به شما داده شده است. خروجی index نود بعدی (Next) نود خواسته شده است. به دلیل محدودیت‌های زمانی برای طراحی داده تست، ممکن است راه حل‌هایی غیر از پیاده‌سازی Next در درخت BST نیز تست‌ها را در زمان مناسب پاس کند. تنها راه حل قابل قبول پیاده‌سازی تابع Next مطابق اسلایدهای درس می‌باشد.

نمونه خروجی:	نمونه ورودی:
0	15 4 9 4 11 -1 8 -1 12 -1 -1 -1 -1 -1 -1 -1 -1
2	15 0 7 2 11 -1 5 -1 12 -1 -1 -1 -1 -1 -1 -1 -1

سوال ۲ – Hash Flooding Attack

ورودی و خروجی:

در جلسه ۲۲م در مورد حمله Hash Flooding توضیح دادیم. شرکت مایکروسافت برای مقابله با این مشکل امنیتی تنظیم UseRandomizedStringHashAlgorithm را برای غیر قابل پیش‌بینی کردن collisionها در پیاده‌سازی HashTableها معرفی کرد. فرض می‌کنیم یکی از وب‌سرویس‌های دشمن[۱] که این تنظیم را فعال نکرده، رشته[۲]های موجود در Web Service Request مشخصی را درون یک HashTable اضافه می‌کند. همچنین فرض کنیم که ظرفیت پیش‌فرض این HashTable را می‌دانیم. به منظور انجام حمله DOS[3] بر اساس Hash Flooding لازم است زمان Lookup در HashTable را به حداکثر برسانیم. برای رسیدن به این هدف لازم است رشته‌های حرفی را انتخاب کنیم که هنگام اضافه شدن به HashTable در یک Bucket قرار بگیرند. همچنین می‌دانیم که با توجه به پویا/Dynamic بودن HashTable چنانچه تعداد عناصر اضافه شده به HashTable از MaxLoadingFactor بیشتر شود، HashTable رشد کرده و تعداد Bucketها تغییر کرده و collisionها نیز تغییر می‌کند. لذا حداکثر تعداد رشته‌هایی که می‌توانیم استفاده کنیم به اندازه MaxLoadingFactor ضرب‌در ظرفیت اولیه HashTable می‌باشد. در این سوال به شما ظرفیت اولیه HashTable به عنوان ورودی داده شده است. وظیفه شما درست کردن حداکثر تعداد رشته می‌باشد که درونی یک Bucket قرار گیرند. برای این مساله MaxLoadingFactor را ۰/۹ در نظر بگیرید. خروجی یک آرایه است که شامل آن تعداد مشخص از string هاست که دارای شماره Bucket یکسانی هستند.

پیاده‌سازی این سوال چندان مشکل نیست. هدف از این سوال بیشتر آزمودن درک شما از HashTable و HashCode می‌باشد. لذا چنانچه سوال مقداری ابهام دارد، این طبیعی است. اگر بخواهیم بیشتر توضیح بدهیم دیگر نکته‌ای در سوال برای آزمودن باقی نمی‌ماند.

نمونه خروجی:	نمونه ورودی:
AH, AD, t, n, l, Z, Y, J, I	10

[۱] دشمن را مطابق میل خود برای توجیه این حمله تفسیر/تعبیر کنید.

[۲] string

[3] Denial Of Service

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_Exam2b`

- 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 `E2b.Tests.csproj` اضافه کنید.

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>...</PropertyGroup>

  <ItemGroup>...</ItemGroup>

  <ItemGroup>...</ItemGroup>

  <ItemGroup>
    <Content Include="TestData\**">
      <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
    </Content>
  </ItemGroup>

</Project>
```