



دانشگاه علم و صنعت ایران

دانشکده مهندسی کامپیوتر

طراحی و تحلیل الگوریتم‌ها

تمرین ۵\*

اساتید حل تمرین: هادی شیخی، ملیکا نوبختیان  
تهیه و تنظیم مستند: مریم سادات هاشمی

استاد درس: سید صالح اعتمادی

نیم‌سال دوم ۱۴۰۰-۱۳۹۹

@hadichhh @MeliNo2000	تلگرام
fb_A5	نام شاخه
A5	نام پروژه/پوشه/پول ریکوست
۱۴۰۰/۲/۴	مهلت تحویل

\*تشکر ویژه از اساتید حل تمرین مریم سادات هاشمی، بنفشه کریمیان، مهسا سادات رضوی، امیر خاکپور، سهیل رستگار و علی آلیاسین که در نیم‌سال دوم سال تحصیلی ۹۷-۹۸ نسخه اول این مجموعه تمرین‌ها را تهیه فرمودند.

## توضیحات کلی تمرین

۱. ابتدا مانند تمرین های قبل، یک پروژه به نام A5 بسازید.
۲. کلاس هر سوال را به پروژه‌ی خود اضافه کنید و در قسمت مربوطه کد خود را بنویسید. هر کلاس شامل دو متد اصلی است:
  - متد اول: تابع Solve است که شما باید الگوریتم خود را برای حل سوال در این متد پیاده سازی کنید.
  - متد دوم: تابع Process است که مانند تمرین های قبلی در TestCommon پیاده سازی شده است. بنابراین با خیال راحت سوال را حل کنید و نگران تابع Process نباشید! زیرا تمامی پیاده سازی ها برای شما انجام شده است و نیازی نیست که شما کدی برای آن بنویسید.
۳. اگر برای حل سوالی نیاز به تابع های کمکی دارید؛ می توانید در کلاس مربوط به همان سوال تابع تان را اضافه کنید.  
اکنون که پیاده سازی شما به پایان رسیده است، نوبت به تست برنامه می رسد. مراحل زیر را انجام دهید.
  ۱. یک UnitTest برای پروژه‌ی خود بسازید.
  ۲. فولدر TestData که در ضمیمه همین فایل قرار دارد را به پروژه‌ی تست خود اضافه کنید.
  ۳. فایل GradedTests.cs را به پروژه‌ی تستی که ساخته اید اضافه کنید.

### توجه:

برای اینکه تست شما از بهینه سازی کامپایلر دات نت حداکثر بهره را ببرد زمان تست ها را روی بیلد Release امتحان کنید، در غیر اینصورت ممکن است تست های شما در زمان داده شده پاس نشوند.

```

1 using Microsoft.VisualStudio.TestTools.UnitTesting;
2 using TestCommon;
3
4 namespace A5.Tests
5 {
6     [DeploymentItem("TestData")]
7     [TestClass()]
8     public class GradedTests
9     {
10         // [TestMethod(), Timeout(300)]
11         [TestMethod(), Timeout(800)]
12         public void SolveTest_Q1ConstructTrie()
13         {
14             RunTest(new Q1ConstructTrie("TD1"));
15         }
16
17         // [TestMethod(), Timeout(500)]
18         [TestMethod(), Timeout(1000)]
19         public void SolveTest_Q2MultiplePatternMatching()
20         {
21             RunTest(new Q2MultiplePatternMatching("TD2"));
22         }
23
24
25         // [TestMethod(), Timeout(750)]
26         [TestMethod(), Timeout(1200)]
27         public void SolveTest_Q3GeneralizedMPM()
28         {
29             RunTest(new Q3GeneralizedMPM("TD3"));
30         }
31
32         // [TestMethod(), Timeout(1400)]
33         [TestMethod(), Timeout(2000)]
34         public void SolveTest_Q4SuffixTree()
35         {
36             RunTest(new Q4SuffixTree("TD4"));
37         }
38
39         // [TestMethod(), Timeout(1200)]
40         [TestMethod(), Timeout(2000)]
41         public void SolveTest_Q5ShortestNonSharedSubstring()
42         {
43             RunTest(new Q5ShortestNonSharedSubstring("TD5"));
44         }
45
46
47         public static void RunTest(Processor p)
48         {
49             TestTools.RunLocalTest("A5", p.Process, p.TestDataName, p.Verifier,
50                 VerifyResultWithoutOrder: p.VerifyResultWithoutOrder,
51                 excludedTestCases: p.ExcludedTestCases);
52         }
53     }
54 }

```

# ۱ ساختن درخت (Trie) از روی تعدادی الگو<sup>۱</sup>

یک درخت برای مجموعه‌ای از الگوها بصورت زیر تعریف می‌شود:

- یک درخت تنها یک ریشه دارد.
- هر یال با یک حرف از حروف الفبا برچسب‌گذاری می‌شود.
- یال‌هایی که از یک گره خارج می‌شوند دارای برچسب‌های متمایزاند.
- الگوها از طریق پیمایش یک مسیر از ریشه درخت به سمت برگ‌ها بدست می‌آیند.
- هر مسیر از ریشه درخت تا برگ‌ها نشان‌دهنده‌ی یک الگو است.

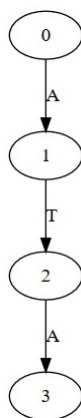
در این سوال شما باید الگوریتمی بنویسید که یک درخت از رشته الگوهای داده شده در ورودی را تشکیل دهد. خط اول فایل ورودی یک عدد صحیح بین ۱ تا ۱۰۰ است که این عدد نشان‌دهنده تعداد الگوهای داده شده برای ساخت درخت است. در هر یک از خطوط بعدی الگوها برای ساخت درخت آورده شده‌اند. طول این رشته الگوها بین ۱ تا ۱۰۰ کاراکتر از بین مجموعه کاراکترهای C G T A است. در این سوال فرض بر آن است که هیچ یک از الگوها زیررشته‌ای از الگوهای دیگر نیستند.

در فایل خروجی ساختار درخت در قالب یک لیست با فرمت مشخص نشان داده شده است. در درختی که با  $n$  گره می‌سازید، گره ریشه را با برچسب صفر شماره‌گذاری کنید و باقی‌گه‌ها را متناسب با ترتیب رشته‌ها در ورودی با اعداد ۱ تا  $n-1$  شماره‌گذاری کنید. هر خط از فایل خروجی نشان‌دهنده یالی از درخت ساخته شده است. هر یال را با یک سه تایی به فرم  $u \rightarrow v : c$  نشان می‌دهیم. دو عدد اول یعنی  $u$  و  $v$  نشان‌دهنده شماره گره مبدا و مقصد یال است و کاراکتر سوم یعنی  $c$  نشان‌دهنده برچسب یال مورد نظر است.

**توجه:**

**دقت کنید که ترتیب نشان دادن یال‌ها در خروجی، متناسب با ترتیب رشته‌های الگو در ورودی است.**

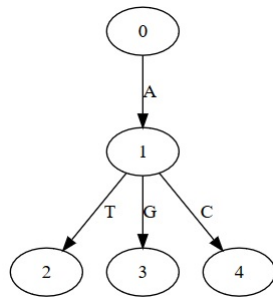
ورودی نمونه	خروجی نمونه
1 ATA	0->1:A 2->3:A 1->2:T



شکل ۱: نمونه اول

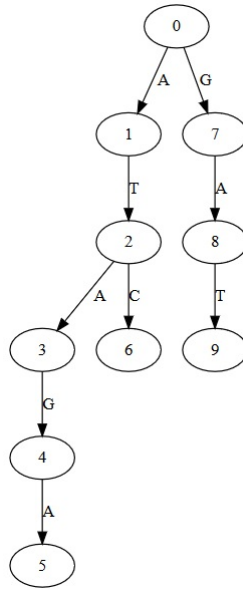
<sup>۱</sup>Construct a Trie from a Collection of Patterns

ورودی نمونه	خروجی نمونه
3 AT AG AC	0->1:A 1->4:C 1->3:G 1->2:T



شکل ۲: نمونه دوم

ورودی نمونه	خروجی نمونه
3 ATAGA ATC GAT	0->1:A 1->2:T 2->3:A 3->4:G 4->5:A 2->6:C 0->7:G 7->8:A 8->9:T



شکل ۳: نمونه سوم

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using TestCommon;
7
8 namespace A5
9 {
10     public class Q1ConstructTrie : Processor
11     {
12         public Q1ConstructTrie(string testDataName) : base(testDataName)
13         {
14             this.VerifyResultWithoutOrder = true;
15         }
16
17         public override string Process(string inStr) =>
18             TestTools.Process(inStr, (Func<long, String[], String[]>) Solve);
19
20         public string[] Solve(long n, string[] patterns)
21         {
22             throw new NotImplementedException();
23         }
24     }
25 }

```

## ۲ کاربرد الگوهای چندگانه ۲

در این سوال باید الگوریتمی بنویسید که محل تعدادی الگو به عنوان زیررشته را در یک متن برگرداند. در خط اول از فایل ورودی یک رشته از متن مورد نظر با طولی بین ۱ تا ۱۰۰۰۰ کاراکتر متشکل از C G T A داده شده است.

در خط دوم یک عدد صحیح بین ۱ تا ۵۰۰۰ که نشان‌دهنده تعداد الگوها است و در هر یک از خطوط بعدی رشته الگوهای داده شده برای یافتن در متن داده شده است. در این سوال فرض بر آن است که هیچ یک از الگوها زیررشته‌ای از الگوهای دیگر نیست.

در فایل خروجی تمامی index هایی از متن ( بصورت صعودی ) که در آن یک رشته از الگوها، به عنوان زیررشته یافت می‌شود وجود دارد ( گذاری index روی متن از صفر شروع می‌شود). در صورتی که پاسخی نیافت مقدار ۱- را برگرداند.

ورودی نمونه	خروجی نمونه
AAA 1 AA	0 1

ورودی نمونه	خروجی نمونه
AA 1 T	-1

ورودی نمونه	خروجی نمونه
AATCGGGTTCAATCGGGGT 2 ATCG GGGT	1 4 11 15

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using TestCommon;
7
8 namespace A5
9 {
10     public class Q2MultiplePatternMatching : Processor
11     {
12         public Q2MultiplePatternMatching(string testDataName) : base(testDataName)
13         {
14         }
15
16         public override string Process(string inStr) =>
17             TestTools.Process(inStr, (Func<String, long, String[], long[]>)Solve);
18
19         public long[] Solve(string text, long n, string[] patterns)
20         {
21             throw new NotImplementedException();
22         }
23     }
24 }

```

### ۳ تطبیق الگوهای چندگانه‌ی تعمیم یافته<sup>۳</sup>

در این سوال راه‌حل سوال قبل تعمیم داده می‌شود به طوری که شما باید الگوریتمی بنویسید که حالتی را که یک الگوی ورودی زیررشته الگوی دیگری باشد را نیز شامل شود. در این حالت رشته‌هایی از الگوها وجود دارند که در طی مسیری از ریشه درخت تا یک گره داخلی (و نه لزوماً برگ) ساخته می‌شوند. در خط اول فایل ورودی یک رشته از متن مورد نظر با طولی بین ۱ تا ۱۰۰۰۰ کاراکتر متشکل از C G T A داده شده‌است. در خط دوم یک عدد صحیح بین ۱ تا ۵۰۰۰ که نشان‌دهنده تعداد الگوها است و در هر یک از خطوط بعدی رشته الگوهای داده شده برای یافتن در متن داده شده است.

**توجه:**

در این سوال فرض بر آن است که الگوها می‌توانند زیررشته‌ای از الگوهای دیگر باشند.

در فایل خروجی تمامی‌هایی index از متن ( بصورت صعودی ) که در آن یک رشته از الگوها بعنوان زیر رشته یافت می‌شود وجود دارد ( گذاری index روی متن از صفر شروع می‌شود). در صورتی که پاسخی نیافت مقدار ۱- را برگرداند.

**توجه:**

اگر بیش از یک الگو در امین i index یافت شود، تنها یکبار i را در خروجی نشان دهید.

ورودی نمونه	خروجی نمونه
AAA 1 AA	0 1

ورودی نمونه	خروجی نمونه
ACATA 3 AT A AG	0 2 -1

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using TestCommon;
7
8 namespace A5
9 {
10     public class Q3GeneralizedMPM : Processor
11     {
12         public Q3GeneralizedMPM(string testDataName) : base(testDataName)
13         {

```



```

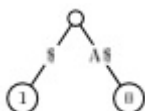
۱۴     }
۱۵
۱۶     public override string Process(string inStr) =>
۱۷     TestTools.Process(inStr, (Func<String, long, String[], long[]>)Solve);
۱۸
۱۹     public long[] Solve(string text, long n, string[] patterns)
۲۰     {
۲۱         throw new NotImplementedException();
۲۲     }
۲۳ }
۲۴ }

```

## ۴ ساختن suffix tree برای یک رشته ۴

درخت‌های پسوندی یک ساختار داده قدرتمند است که در زمینه‌هایی مثل pattern matching و compression کاربرد دارد. هدف از این سوال پیاده‌سازی این داده ساختار است. در این سوال باید الگوریتمی بنویسید که درخت پسوندی از یک رشته ورودی تشکیل دهد. در خط اول فایل ورودی یک رشته که با نماد \$ پایان می‌یابد وجود دارد. این رشته با طولی بین ۱ تا ۵۰۰۰ از مجموعه کاراکترهای C G T A ساخته شده است. در هر یک از خط‌های فایل خروجی یک رشته که نشان‌دهنده‌ی برچسب یالی از درخت است وجود دارد. ترتیب نشان دادن یال‌ها در خروجی باید مطابق با پیمایش درخت بصورت اول عمق (dfs) باشد.

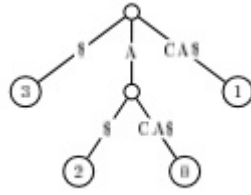
ورودی نمونه	خروجی نمونه
A\$	A\$ \$



شکل ۴: نمونه اول

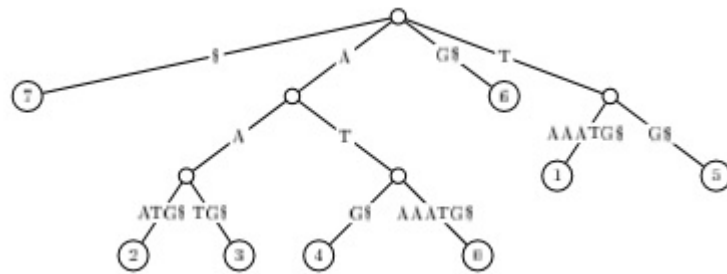
ورودی نمونه	خروجی نمونه
ACA\$	\$ A \$ CA\$ CA\$

Construct the Suffix Tree of a String<sup>۴</sup>



شکل ۵: نمونه دوم

ورودی نمونه	خروجی نمونه
ATAAATG\$	AAATG\$ G\$ A ATG\$ TG\$ A A AAATG\$ T G\$ G\$ \$



شکل ۶: نمونه سوم

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using TestCommon;
7
8 namespace A5
9 {
10     public class Q4SuffixTree : Processor
11     {
12         public Q4SuffixTree(string testDataName) : base(testDataName)
13         {

```

```

۱۴     this.VerifyResultWithoutOrder = true;
۱۵     }
۱۶
۱۷     public override string Process(string inStr) =>
۱۸     TestTools.Process(inStr, (Func<String, String[]>)Solve);
۱۹
۲۰     public string[] Solve(string text)
۲۱     {
۲۲         throw new NotImplementedException();
۲۳     }
۲۴ }
۲۵ }

```

## ۵ پیدا کردن کوچکترین زیر رشته غیر مشترک دو رشته ۵

در این سوال دو رشته به شما داده می‌شود و هدف پیدا کردن کوچکترین زیر رشته غیر مشترک بین آن دو رشته است. شما باید الگوریتمی بنویسید که کوچکترین زیررشته از رشته اول، که در رشته دوم وجود ندارد را پیدا کند. در دو خط اول فایل ورودی ۲ رشته ورودی وجود دارد. رشته‌های ورودی با طولی بین ۱ تا ۲۰۰۰ متشکل از A C G T ساخته شده‌اند. طول این رشته‌ها با هم برابراند. در فایل خروجی کوچکترین زیررشته از رشته اول که در رشته دوم وجود ندارد قرار دارد. ( ممکن است چند جواب متفاوت برای دو رشته وجود داشته باشد که در این صورت باید زیر رشته‌ای که در index کوچکتری وجود دارد را برگردانید.)

ورودی نمونه	خروجی نمونه
A T	A

ورودی نمونه	خروجی نمونه
AAAAAAAAAAAAAAAAAAAA TTTTTTTTTTTTTTTTTTTT	A

ورودی نمونه	خروجی نمونه
CCAAGCTGCTAGAGG CATGCTGGGCTGGCT	CC

ورودی نمونه	خروجی نمونه
ATGCGATGACCTGACTGA CTCAACGTATTGGCCAGA	ATG

Find the Shortest Non-Shared Substring of Two Strings<sup>۵</sup>

```

1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using TestCommon;
8
9 namespace A5
10 {
11     public class Q5ShortestNonSharedSubstring : Processor
12     {
13         public Q5ShortestNonSharedSubstring(string testDataName) : base(testDataName)
14         {
15         }
16
17         public override string Process(string inStr) =>
18             TestTools.Process(inStr, (Func<String, String, String>)Solve);
19
20         private string Solve(string text1, string text2)
21         {
22             throw new NotImplementedException();
23         }
24     }
25 }

```