

# الگوریتم‌های پیشرفته

## تمرین سری دوم

زمان آپلود: 1401/8/13

موعد تحویل: 1401/8/21

مسئول تمرین: مصطفی امیری (mostafa.amiri@ut.ac.ir)



1) معمولاً در پردازش متون طبیعی یکی از مراحل مشترک پیش پردازش، جداکردن کلمات یک جمله از یکدیگر و به اصطلاح توکن‌بندی جمله [1] است. به طور مثال جمله «درس الگوریتم‌های پیشرفته، درسی کاربردی است» را می‌شود به صورت‌های زیر:

درس - الگوریتم‌های - پیشرفته - ، - درسی - کاربردی - است

درس - الگوریتم - های - پیشرفته، - درس - ی - کاربرد - ی - است

درس - الگوریتمها - ی - پیشرفته - ، - درس - ی - کاربرد - ی - است

...

فرض کنید یک لغتنامه از کلمات در اختیار دارید و از شما خواسته شده است که بر اساس آن لغتنامه، تمام حالات توکن‌بندی یک جمله را بدست آورید. الگوریتمی به روش برنامه نویسی پویا طراحی کنید که این کار را انجام دهد.

مرتبه زمانی و فضای حافظه مورد نیاز را بر حسب طول جمله ورودی بدست آورید.

توجه کنید که ممکن است به دلیل اشکالات تایپی فاصله و نیم‌فاصله‌ها در جمله ورودی به درستی رعایت نشده باشد و مثلاً جمله بالا به صورت «درس الگوریتم‌هایپیشرفته، درسیکاربردیاست» تایپ شده باشد.

2) از عبارات منظم [1] برای مشخص کردن الگوهای متنی استفاده می‌شود. دو نماد پر استفاده در این عبارات منظم عبارتند از نماد \* و ؟. نماد \* به معنی صفر یا هر چند کاراکتر دلخواه و نماد ؟ به معنی یک کاراکتر دلخواه است. بنابراین به طور مثال الگوی  $a*b$  می‌تواند بر عباراتی مثل `atab`, `aab`, `ab` تطبیق [2] پیدا کند و عبارت  $a*b?$  نیز میتواند بر عباراتی مثل `abi`, `atabi` تطبیق یابد ولی این الگوی اخیر بر عبارتی مثل `abri` تطبیق پیدا نمی‌کند.

الگوریتمی به روش برنامه نویسی پویا طراحی کنید که یک عبارت منظم با نماد \* و ؟ و یک کلمه را دریافت و تطابق یا عدم تطابق آن را با الگوی داده شده اعلام کند.

مرتبه زمانی و حافظه مورد نیاز این الگوریتم را بدست آورید.

$m \times n$

---

[1] Regular expressions (Regex)

[2] Match

3) یک ماشین تایپ قدیمی فقط قادر است کلمات را با یک اندازه فونت مشخص روی کاغذ چاپ کند. با توجه به این محدودیت تعداد نویسه‌هایی که در یک خط جا می‌شود مقدار ثابت  $M$  است. اگر برای تایپ یک متن شامل تعدادی کلمه با طول‌های (تعداد نویسه‌ها) متفاوت با این ماشین تایپ، هر کلمه را با یک نویسه فاصله از هم جدا کنیم و ملزم باشیم هر کلمه بطور کامل در یک خط باشد، طبیعتاً مجبور خواهیم شد کلمات را در خطوط مختلف تایپ کنیم. برای اینکه شکل کلی متن تایپ شده از زیبایی دیداری مناسبی برخوردار باشد می‌خواهیم که در انتهای خطوط فضای خالی (سفید) زیادی نداشته باشیم. از این رو برای هر خط یک تابع هزینه تعریف می‌کنیم که برابر مکعب تعداد نویسه‌های فاصله در انتهای آن خط است. این هزینه را برای خط آخر برابر صفر در نظر می‌گیریم. بطور مثال اگر کلمات  $i$  تا  $z$  را در یک خط بنویسیم هزینه ایجاد شده به صورت زیر خواهد بود:

$$line\_cost(i,j) = \begin{cases} \infty & \text{اگر این کلمات در یک خط نشوند} \\ 0 & \text{برای خط آخر} \\ \left( M - j + i - \sum_{k=i}^j length(w_k) \right)^3 & \end{cases}$$

هزینه کل در نظر گرفته شده برای یک متن تایپ شده برابر مجموع هزینه سطرهای آن متن است و یک پاسخ بهینه برای شکستن متن به خطوط، پاسخی است که حداقل هزینه را داشته باشد.

الف- نشان دهید که این مساله دارای ساختار sub-optimal هست.

ب- تابع هزینه پاسخ بهینه را به صورت بازگشتی تعریف کنید.

ج- یک الگوریتم بهینه برای محاسبه پاسخ بهینه و هزینه آن ارائه کنید و زمان اجرا و فضای حافظه مورد نیاز را بدست آورید.

4) شما در یک مسابقه شرکت کرده اید که در آن  $n$  سوال وجود دارد و برای هر سوال هم مبلغی  $(d_i)$  جایزه در نظر گرفته شده است. اولین سوالی را که نتوانید پاسخ دهید، مسابقه پایان می پذیرد. اگر احتمال عدم پاسخگویی به هر سوال  $(p_i)$  مشخص باشد، الگوریتمی حریصانه طراحی کنید که بهترین ترتیب انتخاب سوالها برای حداکثر کردن میانگین جایزه مورد انتظار را مشخص کند.

5) فرض کنید تعدادی نقطه به صورت صعودی روی محور افقی داریم. الگوریتمی حریصانه طراحی کنید که کمترین تعداد بازه به طول ۲ را چنان مشخص کند که شامل تمام نقاط داده شده بشوند. به طور مثال نقاط  $\{۱۰.۲ و ۹.۱ و ۸.۸ و ۵.۷ و ۲.۱ و ۲ و ۱.۵\}$  را با حداقل سه بازه به طول ۲ می شود پوشش داد که یک نمونه آن می تواند بازه های  $[1.5, 3.5] - [4, 6] - [8.7, 10.7]$  باشد.