



۱. با استفاده از دو پشته، یک صف پیاده‌سازی کنید که اعمالش در کمترین زمان ممکن انجام شوند.

۲. الف) با کمترین صف، یک پشته را پیاده‌سازی کنید. (بهترین پیچیدگی زمانی با این تعداد صف مدنظر است)
ب) با بهترین پیچیدگی زمانی و تعدادی صف، یک پشته پیاده‌سازی کنید. (در پیچیدگی زمانی بهینه، کمترین صف را استفاده کنید)

۳. مراحل محاسبه‌ی عبارت ریاضی infix و prefix را با حداقل پشته برای هرکدام توضیح دهید.

۴. الف) برای هرکدام از قسمت‌های زیر درختی را بکشید که دو نمایش داده شده برای آن درست باشد. در هر کدام از قسمت‌ها چند درخت صدق می‌کند؟

a) Pre-order: A B D E H C F G I J
Post-order: D H E B F I J G C A

b) In-order: D B H E A F C I G J
Post-order: D H E B F I J G C A

c) Pre-order: A B D E H C F G I J
In-order: D B H E A F C I G J

ب) برای اینکه بتوان یک درخت را به درستی کشید، به حداقل چند پیمایش و کدام پیمایش‌ها از آن احتیاج است؟

۵. در پیاده‌سازی پشته با یک آرایه مشکلی که وجود دارد محدود بودن طول پشته است، با اختیار کردن آرایه‌ها و آزاد کردن آنها پشته‌ای با طول نامحدود بسازید، توجه کنید که پشته ساخته شده باید از لحاظ زمان بهترین پیچیدگی را داشته باشد.

۶. پشته‌ای با سه تابع تعریف شده‌ی push، pop و print داریم که توابع print و pop باید روی پشته‌ای حاوی حداقل یک عنصر اجرا شوند. تابع print عنصر بالای پشته را چاپ می‌کند. با اجرای جایگشتی از این سه تابع می‌توان رشته‌ای را چاپ کرد. مثلاً اگر رشته‌ی S برابر foo باشد ترتیب تابع‌ها می‌تواند به شکل زیر باشد:

push f, print, pop, push o, print, pop, push o, print, pop

در نتیجه برای رشته‌ای به طول n با اجرای دقیقاً 3n تابع می‌توانیم آن را چاپ کنیم.

هدف ما این است که از پشته‌ی خالی شروع کنیم و با کمترین تعداد اجرای این سه تابع رشته‌ی مورد نظر S را خروجی

دهیم. در انتها نیز پشته باید خالی شود. مثلاً رشته‌ی را foo می‌توانستیم با اجرای ۷ تابع چاپ کنیم:

push f, print, pop, push o, print, print, pop

مسئله‌ی ما پیدا کردن کمترین تعداد تابع مورد نیاز برای چاپ رشته است.

الف) مسئله را در حالتی که کاراکترهای S حروف a تا z باشند با کمترین پیچیدگی زمانی حل کنید.

ب) مسئله را در حالتی که کاراکترهای S سه حرف A و B و C باشند با کمترین پیچیدگی زمانی حل کنید.

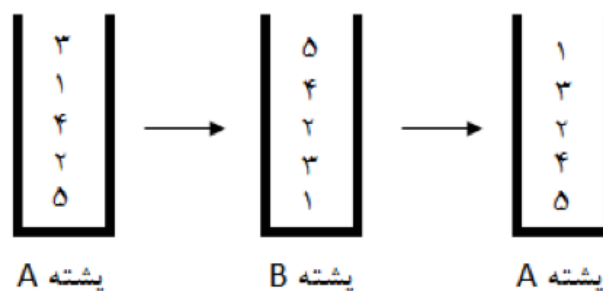
در هیچ کدام از دو حالت نیاز به اثبات بهینه بودن پیچیدگی زمانی نیست، اما باید بهترین الگوریتم از نظر پیچیدگی زمانی ارائه شود و پیچیدگی زمانی محاسبه شود.

۷. مرتب‌ساز پشته‌ای یک مرتب‌ساز با دو پشته است. در ابتدا در پشته اول که آن را پشته‌ی A می‌نامیم اعداد 1 تا n با ترتیبی دلخواه قرار دارند و پشته دوم با نام B خالی است. این مرتب‌ساز قادر است عملیات زیر را انجام دهد:

- در هر مرحله دو عدد بالای پشته‌ی A را در نظر می‌گیرد و عدد کوچکتر را به پشته‌ی B انتقال می‌دهد و این کار را آنقدر تکرار می‌کند که در پشته‌ی A تنها یک عنصر باقی بماند و آن را نیز به پشته‌ی B منتقل می‌کند. سپس اعداد پشته‌ی B را به پشته‌ی A انتقال می‌دهد (توجه کنید که چون A و B پشته هستند ترتیب عناصر برعکس می‌شود).

اگر مرتب‌ساز پشته‌ای عملیات فوق را $1 \leq k \leq n$ بار انجام دهد به ازای چند جایگشت اولیه از اعداد 1 تا n درون A ، در نهایت اعداد بصورت مرتب شده در پشته‌ی A قرار خواهند گرفت؟ (عدد 1 در بالای پشته و عدد n در پایین پشته) جواب را بر حسب n و k محاسبه و اثبات کنید.

بعنوان مثال در شکل زیر وضعیت پشته‌ی A بعد از یک بار انجام عملیات نمایش داده شده است. در این شکل سه گام مشخص شده است که به ترتیب عبارتند از: وضعیت اولیه پشته‌ی A ، نحوه قرار گرفتن اعداد در پشته‌ی B ، وضعیت اعداد در پشته‌ی A بعد از عملیات.



نحوه‌ی تحویل:

لطفاً تمرین را به صورت اسکن شده در یک فایل فشرده با نام HW2[SID].zip در سایت درس آپلود کنید. SID پنج رقم آخر شماره‌ی دانشجویی شما است. یعنی اگر شماره دانشجوییتان ۸۱۰۱۹۲۰۰۰ است، نام فایل‌تان باید HW292000.zip باشد.

نکات پایانی:

- ✓ به ازای هر روز تاخیر ۱۰ درصد از نمره‌ی تمرین را از دست خواهید داد. همچنین بیشترین میزان تأخیر مجاز ۵ روز است.
- ✓ در صورت مشاهده‌ی هرگونه تشابه نمره‌ی هر دو طرف ۱۰۰- منظور می‌گردد و در بار دوم نمره‌ی صفر برای درس منظور می‌گردد.
- ✓ در صورت وجود هر گونه سوال می‌توانید به فروم درس مراجعه کنید. همچنین میتوانید سوالات خود را به ایمیل mehrdad.baxtiari@gmail.com ارسال کنید.