

نمونه سوال با راه حل

ساختمان داده‌ها

دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. پاییز ۱۴۰۱

۱. مقدار برگشتی تابع زیر چیست؟ M یک ماتریس مربعی است. تابع $\text{mult}(A,B)$ حاصلضرب $A \times B$ را برمی‌گرداند.

```
def mat(M):
    P = M
    Q = M
    for i in range(1,n):
        P = mult(P,M)
        Q = mult(Q,P)
    return Q
```

در نهایت Q توانی از ماتریس M خواهد بود. اگر $n = 1$ حلقه اجرا نمی‌شود و مقدار Q برابر با M خواهد بود. بطور کلی اگر $n = i$ مقدار P برابر با M^i خواهد بود. لذا مقدار Q در نهایت برابر با $M^i \times Q$ خواهد شد. در کل برای توان ماتریس M می‌توان رابطه بازگشتی زیر را نوشت.

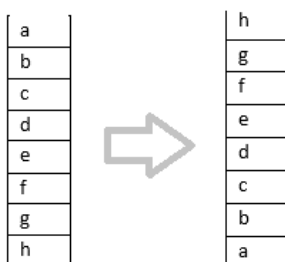
$$T(1) = 1, T(i) = i + T(i - 1)$$

لذا $T(n) = n(n + 1)/2$. پس خروجی تابع بالا

$$Q^{n(n+1)/2}$$

خواهد بود.

۲. وارون کردن یک پشته: می‌خواهیم جای عناصر یک پشته عوض شود بطوریکه عناصر به ترتیب وارون در پشته قرار گیرند. یک نمونه در شکل زیر نشان داده شده است.



این کار را می‌خواهیم با داشتن محدودیتهای مختلف انجام دهیم. برای هر یکی از موارد زیر یک الگوریتم ارائه دهید و زمان اجرای آن را ذکر کنید. توجه کنید در ساختار داده‌های مورد استفاده تنها می‌توانیم از اعمال اصلی مربوط آنها استفاده کنیم (برای مثال نمی‌توانیم جای دو پشته را تعویض کنیم و یا با انجام یک دستور یک پشته را در پشته دیگر کپی کنیم).

(آ) مجاز هستیم از پشته‌ای دیگر و به اندازه $O(1)$ حافظه اضافی استفاده کنیم.

فرض کنید پشته اصلی $S1$ و پشته کمکی $S2$ است. اگر عناصر را یکی یکی از بالای پشته $S1$ حذف کنیم و در پشته $S2$ درج کنیم، در نهایت پشته $S2$ وارون پشته $S1$ خواهد شد. اما مجاز نیستیم از دستورالعمل $S1 = S2$ و یا کپی کردن استفاده کنیم. برای رسیدن به هدف، مثل مرتب سازی انتخابی، در هر مرحله یکی از عناصر پشته $S1$ را به جای درست خود منتقل می‌کنیم. برای نمونه در مثال بالا اول a را به کف پشته $S1$ می‌بریم. برای انجام

این کار ابتدا عنصر a را در جایی از حافظه نگه می‌داریم و عناصر زیر آن را به پشته کمکی $S2$ منتقل می‌کنیم. حال a را در کف پشته خالی $S1$ درج می‌کنیم و دیگر کاری به آن نداریم. حال عناصر پشته $S2$ را به $S1$ برمی‌گردانیم. دوباره همین کار را تکرار می‌کنیم تا عنصر بعدی (b) بالای a قرار گیرد. به همین ترتیب بقیه عناصر را در جای درست خود درج می‌کنیم. زمان اجرا متناسب با $O(n^2)$ خواهد بود.

(ب) مجاز هستیم از دو پشته دیگر و به اندازه $O(1)$ حافظه اضافی استفاده کنیم.

فرض کنید پشته اصلی $S1$ و پشته های کمکی $S2$ و $S3$ باشند. عناصر را یکی یکی از بالای پشته $S1$ حذف می‌کنیم و در پشته $S2$ درج می‌کنیم. پشته $S2$ وارون پشته $S1$ خواهد شد. حال عناصر را یکی یکی از پشته $S2$ به $S3$ منتقل می‌کنیم. در نهایت، عناصر را از $S3$ به $S1$ منتقل می‌کنیم. پشته $S1$ وارون شده است. زمان اجرا متناسب با $O(n)$ است.

(ج) می‌توانیم از یک صف و به اندازه $O(1)$ حافظه اضافی استفاده کنیم.

کافی است عناصر را یکی یکی از $S1$ حذف و در صف Q درج کنیم. پس از اتمام کار آنها را یکی یکی از صف Q خارج کرده و به پشته $S1$ برمی‌گردانیم. حال پشته Q وارون شده است. زمان اجرا متناسب با $O(n)$ است.

(د) می‌توانیم از یک هرم بیشینه و به اندازه $O(1)$ حافظه اضافی استفاده کنیم.

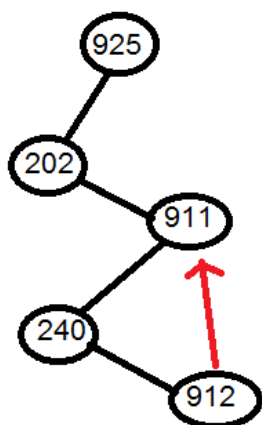
می‌توانیم به هر عنصر در پشته یک کلید عددی منحصر بفرد بدهیم. مثلاً بالای پشته عدد n ، عنصر بعدی $n-1$ و به همین ترتیب عنصر کف پشته کلیدش 1 است. عناصر را از پشته خارج کرده و در هرم بیشینه درج می‌کنیم. این کار با هزینه $O(n \log n)$ امکان پذیر است. حال هر بار عنصر ماکزیمم را از هرم بیشینه حذف می‌کنیم و در بالای پشته درج می‌کنیم. در کل زمان اجرا $O(n \log n)$ خواهد بود.

۳. تابع زیر ریشه یک درخت باینری را دریافت می‌کند. این تابع چه چیزی را محاسبه می‌کند؟

```
def func(TreeNode r):
    if T is None: return 0
    else:
        m1 = 1 + func(r.left)
        m2 = 1 + func(r.right)
        if (m1 > m2): return m1
        else: return m2
    return Q
```

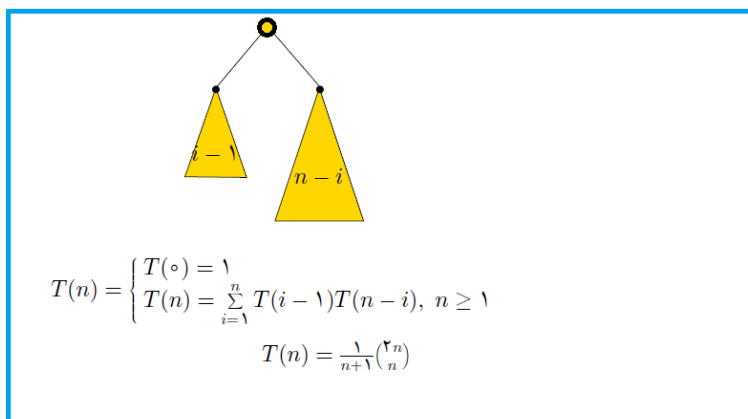
ارتفاع درخت را محاسبه می‌کند.

۵. فرض کنید در یک BST دنبال عدد 363 می‌گردیم. کدام یک از دنباله‌های زیر نمی‌تواند دنباله جستجوی 363 در این ساختار داده باشد؟

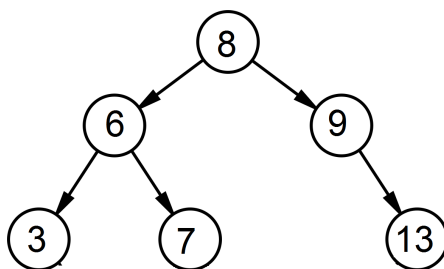


- (a) 2, 252, 401, 398, 330, 344, 397, 363
- (b) 924, 220, 911, 244, 898, 258, 362, 363
- (c) 2, 399, 387, 219, 266, 382, 381, 278, 363
- ✗ (d) 925, 202, 911, 240, 912, 245, 363
- (e) 935, 278, 347, 621, 399, 392, 358, 363

۶. چند درخت باینری متفاوت با n عدد متمایز می‌توان ساخت؟ فرض کنید $T(n)$ تعداد چنین درختهایی باشد. یک رابطه بازگشتی برای $T(n)$ بدست آورید.



۷. چهار ترتیب مختلف از مجموعه اعداد $\{3, 6, 7, 8, 9, 13\}$ را بنویسید که درج آنها در یک BST درخت زیر را نتیجه دهد.



- 8, 6, 3, 7, 9, 13
- 8, 6, 7, 3, 9, 13
- 8, 9, 13, 6, 7, 3
- 8, 9, 13, 6, 3, 7

۸. چند ترتیب مختلف از اعداد مذکور باعث ایجاد درخت بالا می‌شود؟

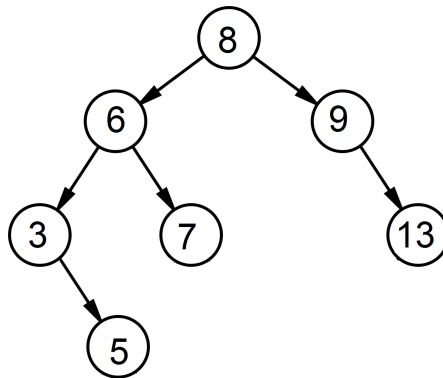
$$2 \times 1 \times \binom{5}{3}$$

۹. ترتیب حاصل از پیمایش preorder درخت بالا را بنویسید.

3, 7, 6, 13, 9, 8

۱۰. به درخت بالا کلید 5 را اضافه کنید و سپس کلید 6 را طبق قاعده‌ای که در جزوه درس پیشنهاد شده حذف کنید. درخت حاصل را رسم کنید.

بعد از درج 5 درخت بصورت زیر در می‌آید.



چون 6 هر دو فرزند چپ و راست را دارد، عنصر کمینه در زیردرخت سمت راست جایگزین آن میشود.

