فصل

۳

داده‌ساختارهای ساده

۱.۳ بله- خیر

۱.۳ n عنصر متمایز و قابل مقایسه داده شده‌اند. می‌خواهیم با این عناصر یک هرم کمینه بسازیم. به‌این منظور، ابتدا یک هرم خالی را در نظر می‌گیریم و سپس این n عنصر را یک‌به‌یک در این هرم درج می‌کنیم. در این‌صورت آیا زمان ساخت هرم از $\mathcal{O}(n \lg n)$ است؟

☐ بله ☐ خیر

۲.۳ فرض کنید هرم کمینه‌ی H با n عنصر و نیز n عنصر نامرتب دیگر به‌طور جداگانه داده شده‌اند. آیا با هزینه‌ای متناسب با $\mathcal{O}(n)$ می‌توان یک هرم از همه‌ی این $2n$ عنصر ساخت؟ فرض کنید که هرم‌های ورودی و نیز هرم نهایی به‌صورت درخت داده شده‌اند و شما امکان ریختن عناصر آن‌ها در یک آرایه (و انجام اعمال مربوط به آن) را ندارید.

☐ بله ☐ خیر

۳.۳ آیا می‌توان یک هرم با n عنصر را در زمان $\mathcal{O}(n)$ به یک د.د.ج[1] تبدیل کرد؟

☐ بله ☐ خیر

۴.۳ k هرم که اندازه‌ی هر کدام n است داده شده‌اند. آیا می‌توان در زمان $\mathcal{O}(k \lg n)$، با عناصر این هرم‌ها، یک هرم جدید به اندازه‌ی nk ساخت؟ فرض کنید $k < n$ و هرم‌ها به‌صورت درختی (با اشاره‌گر) داده شده و به‌همین ساختار هم در خروجی مورد نیاز هستند.

☐ بله ☐ خیر

[1] درخت دودویی جست‌وجو (binary search tree)

⋆ **۵.۳** یک لیست پیوندی یک‌سویه‌ی بزرگ داریم. می‌خواهیم بررسی کنیم که آیا آخرین اشاره‌گر این لیست تهی است یا این‌که به یکی دیگر از عناصر لیست اشاره می‌کند. دقت کنید که فقط یکی از این دو حالت ممکن است و ما از طول لیست اطلاعی نداریم.
با فرض آن‌که تعداد عناصر این لیست n است و ما فقط به عنصر اول لیست دست‌رسی داریم، آیا الگوریتمی از زمان $\mathcal{O}(n)$ و حافظه‌ی اضافی $\mathcal{O}(1)$ برای تشخیص این امر وجود دارد؟

☐ بله ☐ خیر

۶.۳ با توجه به تعریف ماتریس مانگ در مسئله‌ی ۹۲.۳، آیا ماتریس زیر مانگ است؟

۱۰	۱۳	۲۸	۲۵
۱۱	۱۲	۲۵	۲۲
۴۰	۳۲	۳۷	۳۳
۱۳	۴	۱۰	۷
۹۶	۸۲	۸۸	۸۴

☐ بله ☐ خیر

۲.۳ لیست، پشته و صف

۷.۳ عمل درج یک عنصر در یک لیست پیوندی دوسویه، چه‌تعداد از اشاره‌گرهای *next* و *prev* را تغییر می‌دهد؟

۱ ☐ یک *next* و یک *prev*
۲ ☐ یک *next* و دو *prev*
۳ ☐ دو *next* و یک *prev*
۴ ☐ دو *next* و دو *prev*

۸.۳ کدام‌یک از اعمال زیر را نمی‌توان به‌صورت کارا بر روی یک لیست پیوندی یک‌سویه اجرا کرد؟

۱ ☐ دست‌رسی به عنصر در موقعیت فعلی
۲ ☐ درج یک عنصر بعد از موقعیت فعلی
۳ ☐ درج یک عنصر در موقعیت فعلی
۴ ☐ دست‌رسی به عنصر بعدی

۹.۳ یک لیست خطی یک‌سویه با دو اشاره‌گر f و r که به‌ترتیب به عنصر اول و آخر لیست اشاره می‌کنند پیاده‌سازی شده است. هزینه‌ی کدام‌یک از اعمال زیر وابسته به تعداد عناصر لیست است؟

۱ ☐ حذف آخرین عنصر
۲ ☐ حذف اولین عنصر
۳ ☐ درج پس از آخرین عنصر
۴ ☐ درج قبل از اولین عنصر

۱۰.۳ رویه‌ی روبه‌رو برای وارون کردن یک لیست پیوندی L پیش‌نهاد شده است (L عنصر اول لیست است). کدام گزینه در مورد این الگوریتم درست است؟

```
REVERSE (L)
1  if L = null or next[L] = null
2     then return L
3     else  R ← REVERSE(next[L])
4           next[next[L]] ← L
5           next[L] ← null; return R
```

۱ همیشه درست کار می‌کند. ۲ همیشه نادرست کار می‌کند.
۳ فقط برای لیست‌ها به‌اندازه‌ی ۱ و ۲ درست کار می‌کند.
۴ فقط برای لیست‌ها به‌اندازه‌ی حداکثر ۳ درست کار می‌کند.

۱۱.۳ رویه‌ی روبه‌رو برای وارون کردن یک لیست L پیش‌نهاد شده است. کدام گزینه در مورد آن درست است؟

```
REVERSE (L)
1  if L = null then return L
2  r ← REVERSE(next[L])
3  next[next[L]] ← L
4  next[L] ← null
5  return r
```

۱ همیشه درست کار می‌کند. ۲ همیشه نادرست کار می‌کند.
۳ فقط برای لیست‌ها به‌اندازه‌ی ۱ و ۲ درست کار می‌کند.
۴ فقط برای لیست‌ها به‌اندازه‌ی حداکثر ۳ همواره درست کار می‌کند.

۱۲.۳ رویه‌ی زیر برای کپی کردن لیست پیوندی دوسویه‌ی L و تولید لیست پیوندی دوسویه‌ی LC پیش‌نهاد شده است:

```
COPY (L)
1  if L = null
2     then LC ← null
3     else  LC ← ALLOCATE-NODE(element[L], null, null)
4           next[LC]← COPY(next[L])
5           if next[L] ≠ null then prev[next[LC]] ← LC
6  return LC
```

فرض کنید که اشاره‌گر $next[LC]$ برای مقدار گرفتن نیازی به new شدن (شبیه زبان C++ یا جاوا) ندارد. در این صورت

۱ این الگوریتم همواره درست است. ۲ این الگوریتم هیچ‌وقت درست نیست.
۳ این الگوریتم گاهی اوقات موجب خطای زمان اجرا می‌شود.
۴ این الگوریتم فقط برای لیست‌های به طول کم‌تر از ۳ درست است.

۱۳.۳ بر روی پشته‌ای به نام S اعمال زیر را می‌توان انجام داد:

PUSH (S, x): x را در بالای S درج کن، هزینه‌ی هر عمل $\mathcal{O}(1)$ است.

POP (S): عنصر بالای S را حذف کن، هزینه‌ی هر عمل $\mathcal{O}(1)$ است.

MULTIPOP (S, k): اگر $k \leq$ SIZE (S) باشد، k عنصر بالای پشته و گرنه همه‌ی عناصر پشته را یک‌به‌یک حذف کن (SIZE (S) تعداد عناصر پشته است). هزینه‌ی این کار $\mathcal{O}(\min\{k, \text{SIZE}(S)\})$ است.

ترتیبی از n عمل فوق را با ترکیب دل‌خواه و پارامتر k درست و دل‌خواه برای هر عمل MULTIPOP را بر روی پشته‌ی S که در ابتدا تهی است انجام می‌دهیم. مجموع هزینه‌های این اعمال در بدترین حالت چقدر است؟

۱ $\Theta(n)$ ۲ $\Theta(n \lg n)$ ۳ $\Theta(n^2)$ ۴ $\Theta(n * \min\{k, n\})$

۱۴.۳ در ورودی یک پشته اعداد ۱ تا n به ترتیب قرار دارند (۱ در ابتدای ورودی است). عمل PUSH اولین عدد ورودی را برداشته و در بالای پشته قرار می‌دهد. عمل POP عدد بالای پشته را برداشته و در انتهای دنباله‌ی خروجی می‌نویسد. با ترکیبی مناسب از n عدد PUSH و n عدد POP، می‌توان جای‌گشتی از اعداد ۱ تا n را در خروجی تولید کرد که به آن «جای‌گشت قابل تولید» می‌گوییم. برای نمونه برای $n = 4$، جای‌گشت $\langle 2, 3, 1, 4 \rangle$ (که عناصر آن از چپ به راست POP شده‌اند) قابل تولید است. برای $n = 8$، کدام‌یک از جای‌گشت‌های زیر قابل تولید نیست؟

۱ $\langle 4, 3, 7, 8, 6, 2, 5, 1 \rangle$ ۲ $\langle 8, 7, 6, 5, 4, 3, 2, 1 \rangle$
۳ $\langle 4, 3, 2, 1, 8, 7, 6, 5 \rangle$ ۴ $\langle 5, 4, 7, 8, 6, 3, 2, 1 \rangle$

۱۵.۳ هزینه‌ی کدام‌یک از اعمال زیر در یک صف که به ابتدا و انتهای آن اشاره‌گر داریم ثابت نیست؟

۱ درج یک عنصر در انتهای صف ۲ حذف یک عنصر از ابتدای صف
۳ حذف کوچک‌ترین عنصر از صف ۴ دست‌رسی به عنصر انتهای صف

۳.۳ درخت‌ها

⋆ **۱۶.۳** با n عنصر چند درخت دودویی متوازن با ارتفاع $h = \lfloor \lg n \rfloor$ می‌توان ساخت؟

۱ ۱ ۲ $\binom{2^h}{n - 2^h + 1}$ ۳ $\binom{n}{n - 2^h}$ ۴ $\binom{n}{n - 2^{h-1}}$

۱۷.۳ اگر دنباله‌ی SBDHXEJKTFG نتیجه‌ی نمایش پیش‌وندی یک درخت دودویی کامل باشد، کدام گزینه صحیح است؟

یک درخت دودویی کامل درختی مانند هرم است که حداکثر اختلاف ارتفاع برگ‌های آن یک است و برگ‌های پایین‌ترین عمق از سمت چپ چیده شده باشند. (ارشد ۱۳۸۶)

۱ HXDBJKEFGTS پیمایش پس‌وندی و HDXBEJKSTFG پیمایش میان‌وندی است.
۲ HXDJKEBFGTS پیمایش پس‌وندی و HDXBEJKSTFG پیمایش میان‌وندی است.
۳ HXDBJKEFGTS پیمایش پس‌وندی و HDXBJEKSFTG پیمایش میان‌وندی است.
۴ HXDJKEBFGTS پیمایش پس‌وندی و HDXBJEKSFTG پیمایش میان‌وندی است.

۱۸.۳ یک درخت دودویی که هر گره آن دو فرزند دارد را در نظر بگیرید، که برچسب هر گره آن یک حرف انگلیسی است و برچسب‌ها مجزا هستند. اگر ترتیب‌های پیش‌وندی و پس‌وندی این درخت (از چپ‌به‌راست) به‌صورت زیر باشند:

پیش‌وندی	A	B	D	J	E	O	P	F	C	G	M	H	I	K	L
پس‌وندی	D	O	P	E	F	J	B	G	H	K	L	I	M	C	A

ترتیب میان‌وندی این درخت کدام است؟ (ارشد ۱۳۹۱)

۱ DBOEPFJAGCHMKIL
۲ DBOEPJFAGCHIKML
۳ DBOEPJFACGHMKIL
۴ DBOEPJFAGCHMKIL

۱۹.۳ چندتا از گزاره‌های زیر در مورد ارتفاع یک گره (یا درخت) درست است؟

- ارتفاع یک گره یک واحد کم‌تر از ارتفاع پدرش است.
- ارتفاع درخت تهی ۱− است.
- ارتفاع برگ صفر است.
- همیشه یک گره وجود دارد که ارتفاع و عمق آن برابر باشد.

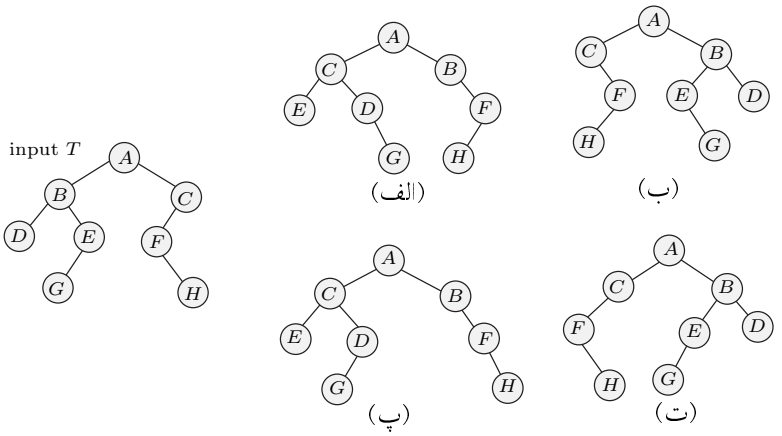
۱ ۱ ۲ ۲ ۳ ۳ ۴ ۴

۲۰.۳ در پیاده‌سازی درخت دودویی معادل، یا «چپ‌ترین فرزند- برادر سمت راست» درخت، کدام‌یک از گزینه‌های زیر درست است؟

۱ دست‌رسی به پدر همیشه ساده است.
۲ برای درخت‌های دودویی مناسب است.
۳ تعداد اشاره‌گرهای هر گره برابر تعداد فرزندان آن گره در درخت اصلی است.
۴ هیچ‌کدام از گزینه‌های فوق همیشه درست نیستند.

۲۱.۳ رویه‌ی روبه‌رو داده شده است. کدام‌یک از درخت‌های (الف) تا (ت) خروجی اجرای این رویه بر روی درخت ورودی T در همین شکل است؟

```
XTree (T)
1  if T = null
2     then return null
3     else  New-Node(p)
4           element[p] ← element[T]
5           left[p] ← XTree(right[T])
6           right[p] ← XTree(left[T])
7           return p
```



۱ الف ۲ ب ۳ پ ۴ ت

۲۲.۳ فرض کنید که $\text{LeafC}(T)$ تعداد برگ‌های درخت دودویی T و اگر T تهی باشد صفر را برمی‌گرداند. هم‌چنین، فرض کنید تابع $\text{IsLeaf}(T)$ اگر T برگ باشد مقدار ۱ وگرنه مقدار صفر را برمی‌گرداند. کدام‌یک از رابطه‌های بازگشتی زیر درست است؟ (ارشد ۱۳۹۱)

۱ $\text{LeafC}(T) = \text{LeafC}(Left[T]) + \text{LeafC}(Right[T])$

۲ $\text{LeafC}(T) = \text{LeafC}(left[T]) + \text{LeafC}(right[T]) + 1$

۳ $\text{LeafC}(T) = \text{LeafC}(left[T]) + \text{LeafC}(right[T]) + \text{IsLeaf}(T)$

۴ $\text{LeafC}(T) = \text{LeafC}(left[T]) + \text{LeafC}(right[T]) + \text{IsLeaf}(T) + 1$

۲۳.۳ یک درخت ۲- کامل درختی است که هر گره آن صفر یا ۲ فرزند دارد. اگر $n(h)$ و $N(h)$ به‌ترتیب بیشینه و کمینه‌ی تعداد گره‌های یک درخت ۲- کامل به ارتفاع h باشد، برای $h > 0$، مقادیر $n(h)$ و $N(h)$ به‌ترتیب کدام‌اند؟

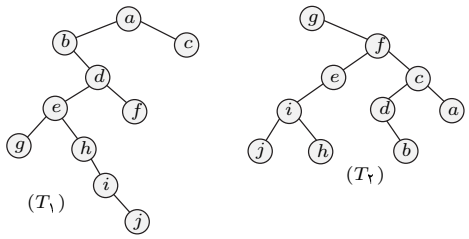
۱ $h + 1$ و 2^{h-1}

۲ $h + 1$ و 2^{h+1}

۳ $2h + 1$ و $2^h - 1$

۴ $2h + 1$ و $2^{h+1} - 1$

۲۴.۳ درخت‌های T_1 و T_2 مطابق شکل زیر داده شده‌اند.



کدام‌یک از روش‌های پیمایش به‌ترتیب بر روی T_1 و T_2 دنباله‌ی یک‌سانی از برچسب‌های گره‌ها را تولید می‌کند؟

۱ پیش‌ترتیب و پس‌ترتیب
۲ پس‌ترتیب و میان‌ترتیب
۳ پس‌ترتیب و پس‌ترتیب
۴ هیچ‌کدام

۲۵.۳ فرض کنید T یک درخت ۲- کامل باشد (تعداد فرزندان هر گره صفر یا ۲ است). هم‌چنین فرض کنید

- $n(T)$ تعداد گره‌های غیر برگ در T،
- $h(T)$ ارتفاع T (فاصله‌ی دورترین برگ از ریشه‌ی T) و
- T_R و T_L به‌ترتیب زیردرخت‌های راست و چپ T باشند.

اگر تابع F به‌صورت زیر تعریف شده باشد:

$$F(T) = \begin{cases} 0 & \text{if } T \text{ is a leaf} \\ F(T_L) + F(T_R) + \min\{h(T_L), h(T_R)\} & \text{otherwise} \end{cases}$$

آن‌گاه، $F(T)$ برابر است با:

۱ $n(T) + h(T)$
۲ $n(T) + h(T) + ۱$
۳ $n(T) - h(T)$
۴ $n(T) - h(T) - ۱$

۲۶.۳ چندتا از گزاره‌های زیر در مورد یک درخت دودویی با n گره همیشه درست است؟

- تعداد گره‌های با دو فرزند برابر است با تعداد برگ‌ها منهای ۱.
- تعداد زیردرخت‌ها حداکثر برابر است با $\mathcal{O}(\frac{n(n-۱)}{۲})$.
- می‌توان درختی با ۲۰ رأس ساخت که در آن تعداد فرزندان هر رأس ۲ یا صفر باشد.
- ارتفاع درخت حداکثر برابر است با $\lceil \log_2 n \rceil$.

۱ ۰ ۲ ۱ ۳ ۲ ۴ ۳

⋆ **۲۷.۳** در یک درخت دودویی T، برای هر برگ x واقع در عمق d تابع $w(x) = ۲^{-d}$ را تعریف می‌کنیم. کدام‌یک از گزینه‌های زیر برابر است با مجموع $w(x)$ها برای همه‌ی برگ‌های درخت T، یا $\alpha = \sum_{\text{leaf } x} w(x)$ است؟

۱ $\alpha \leq ۱$ ۲ $\alpha = ۱$ ۳ $\alpha > ۱$ ۴ $\alpha < ۱$

۲۸.۳ حداکثر تعداد گره‌های یک درخت دودویی با b برگ برابر است با:

۱ $۲b - ۱$ ۲ ۲^b
۳ $۲^{b+۱}$ ۴ کران بالا ندارد

۲۹.۳ پیمایش سطح‌ترتیب گره‌های یک درخت را به‌ترتیب سطح آن‌ها و در هر سطح از چپ‌به‌راست ملاقات می‌کند. ترتیب ملاقات برگ‌های یک درخت در پیمایش سطح‌ترتیب برابر کدام روش است؟

۱ میان‌ترتیب ۲ پس‌ترتیب
۳ پیش‌ترتیب ۴ هیچ‌کدام

۳۰.۳ با داشتن کدام زوج از اطلاعات زیر نمی‌توان یک درخت دودویی (جست‌وجو یا عادی) را به‌طور یک‌تا ساخت؟

۱ دنباله‌های میان‌ترتیب و پیش‌ترتیب
۲ دنباله‌های میان‌ترتیب و پس‌ترتیب
۳ دنباله‌های پس‌ترتیب و پیش‌ترتیب
۴ دنباله‌ی میان‌ترتیب و ارتفاع هر عنصر

۳۱.۳ آرایه‌ی n عضوی A و عدد $m < n/۲$ داده شده‌اند. می‌خواهیم درایه‌های آرایه‌ی $mins[۱..n - m + ۱]$ را که به‌صورت زیر تعریف می‌شوند به‌دست آوریم.

$$mins[i] = \min_{j=i}^{i+m-۱} A[i]$$

با چه الگوریتم کارایی می‌توان همه‌ی درایه‌های $mins$ را به‌دست آورد؟ (ارشد ۱۳۸۷)

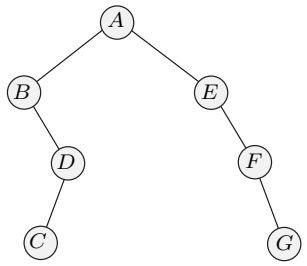
۱ $\mathcal{O}(m \lg n)$ ۲ $\mathcal{O}(n \lg m)$
۳ nm ۴ $(n+m)\lg(n+m)$

۳۲.۳ فرض کنید T یک درخت دودویی کامل با n گره و به ارتفاع $\lg n$ است. می‌خواهیم مسیر موجود از یک رأس u به یک رأس دیگر به نام v را بیابیم. گره‌های u و v داده شده‌اند و می‌دانیم که هر گره از این درخت به گره‌های فرزند و گره پدر دسترسی دارد.

۱ این کار را نمی‌توان در کم‌تر از $\mathcal{O}(n)$ انجام داد.
۲ سریع‌ترین روش استفاده از الگوریتم دایکستراست.
۳ این کار متناسب با ارتفاع درخت و با $\mathcal{O}(\lg n)$ امکان‌پذیر است.
۴ این کار را می‌توان در $\mathcal{O}(\lg^{۲} n)$ انجام داد.

۳۳.۳ خروجی رویه‌ی PrintTree() اگر برروی درخت زیر اجرا شود کدام است؟ فرض کنید Write(s) رشته/عنصر پارامترش (s) را در خروجی می‌نویسد.

```
PrintTree (T)
1  if T = null then return
2  if left[T] ≠ null then Write('(')
3  PrintTree(left[T])
4  if left[T] ≠ null then Write(')')
5  Write(element[T])
6  if right[T] ≠ null then Write('(')
7  PrintTree(right[T])
8  if right[T] ≠ null then Write(')')
```



۱ `A(B(D(C)))(E(F(G)))`
۲ `(B(D(C)))A(E(F(G)))`
۳ `(B((C)D))A(E(F(G)))`
۴ `(C(D(B)))A(G(F(E)))`

۳۴.۳ ⋆ می‌دانیم که در یک درخت دودویی، سطح (یا عمق) یک گره برابر طول مسیر از آن گره تا ریشه است. ارتفاع درخت هم بیش‌ترین سطح گره‌ها در آن درخت است. «پهنای[۲]» یک درخت دودویی T را برابر بیش‌ترین تعداد گره‌های هم‌سطح در T تعریف می‌کنیم. آیا درخت دودویی با n گره و ارتفاع و پهنای زیر وجود دارد؟

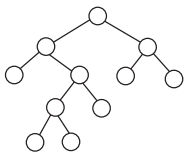
I. ارتفاع $\Theta(n)$ و پهنای ۱
II. ارتفاع $\Theta(\log n)$ و پهنای $\Theta(n)$
III. ارتفاع $\Theta(n)$ و پهنای $\Theta(n)$
IV. ارتفاع $\Theta(\log n)$ و پهنای $\Theta(\sqrt{n})$

چندتا از موارد فوق امکان‌پذیر است؟

۱ ۱ ۲ ۲ ۳ ۳ ۴ ۴

[۲] width

۳۵.۳ به‌چند حالت می‌توان اعداد ۱ تا ۱۱ را در گره‌های درخت روبه‌رو برچسب‌گذاری کرد تا عدد هر گره از اعداد فرزندان آن بزرگ‌تر باشد؟ دقت کنید که از هر ۱۱ عدد باید استفاده شود و تکرار مجاز نیست. (ارشد ۱۳۹۱)



۱ ۹۶ ۲ ۲۰۴۸ ۳ ۱۱۵۲۰ ۴ ۳۶۲۹۸۸۰

۴.۳ عبارت، درخت عبارت

۳۶.۳ در تکه‌برنامه‌ی زیر، دستور `PUSH X` یعنی درج `X` در بالای پشته، دستور `POP X` یعنی حذف عنصر بالای پشته و ذخیره‌ی آن در متغیر `X` است. هم‌چنین عمل‌گر OP می‌تواند `MULT` یا `ADD` به‌ترتیب به‌معنی ضرب و جمع باشد. با فرض آن‌که `A` و `B` دو عنصر بالای پشته هستند، دستور OP این دو عنصر را حذف و عمل OP را بر روی آن دو (یعنی `A` OP `B`) را انجام می‌دهد و حاصل را در بالای پشته درج می‌کند.

مقدار `Z` در انتهای اجرای دستورهای زیر مقدار کدام‌یک از عبارت‌های زیر است؟

```
1.  PUSH A          7.  PUSH T1
2.  PUSH B          8.  PUSH T1
3.  ADD             9.  MULT
4.  POP T1         10.  MULT
5.  PUSH B         11.  ADD
6.  PUSH C         12.  POP Z
```

۱ $B+C(A+B)^2$ ۲ $(B+C)(A+B)^2$
۳ $(A+B)(B^2+C^2)$ ۴ $(B+C^2)(A+B)$

۳۷.۳ با فرض اِعمال اولویت عمل‌گرها، عبارت `a-b*c+d-e/g/h` را به‌چند طریق می‌توان به‌درستی پرانتزگذاری کرد به‌طوری‌که مقدار عبارت حاصل به‌ازای همه‌ی مقادیر a تا h با مقدار عبارت اصلی برابر شود؟

دقت کنید که دور یک متغیرِ تنها در عبارت پرانتز گذاشته نمی‌شود و یک عبارت پرانتزگذاری شده باید صفر یا تعداد زوجی پرانتز داشته باشد. برای نمونه، $((a+b)+c)$، $a+(b+c)$ و $a+b+c$ سه عبارت پرانتزگذاری‌شده‌ی درست برای $a+b+c$ هستند ولی $(a)+b+c$ عبارت درستی برای آن نیست.

۱ ۳۲ ۲ ۶۴ ۳ ۹۶ ۴ ۱۲۸

۳۸.۳ کدام‌یک از داده‌ساختارهای زیر برای تعیین درستی پرانتزگذاری یک عبارت از بقیه مناسب‌تر است؟

۱ جدول درهم‌سازی
۲ صف اولویت
۳ درخت دودویی جست‌وجو
۴ پشته

۳۹.۳ فرض کنید که پشته با آرایه‌ی $S[۱..N]$ و اندیس i پیاده‌سازی شود. اگر متغیرها به‌درستی مقداردهی اولیه شوند و مقدار i در محدوده‌ی تعیین‌شده باشد، دستورهای زیر اعمال PUSH و POP را انجام می‌دهند:

```
PUSH (S, x)
1  S[i] ← x
2  i ← i + 1
```

```
POP (S)
1  i ← i − 1
2  return S[i]
```

کدام‌یک از دستورهای زیر مقداردهی اولیه‌ی درست برای i است؟

۱ $i \leftarrow ۰$
۲ $i \leftarrow ۱$
۳ $i \leftarrow N$
۴ $i \leftarrow N - ۱$

۴۰.۳ این مسئله، ادامه‌ی مسئله‌ی ۳۹.۳ است.

با فرض آن‌که در هر مورد از تغییرهای زیر مقداردهی اولیه به i نیز به‌درستی تغییر می‌کند، با کدام‌یک از تغییرهای زیر در دستورهای داده‌شده‌ی PUSH و POP در مسئله‌ی پیشین، پشته به‌درستی پیاده‌سازی می‌شود؟

a) کدهای PUSH و POP با هم جابه‌جا شوند.
b) کد PUSH مقدار i را یک واحد کم کند و POP آن‌را یک واحد اضافه کند.
c) ترتیب دستورهای هر یک از کدها تغییر کند.

۱ فقط a
۲ فقط b
۳ فقط c
۴ b و c

۴۱.۳ تعداد گره‌های یک درخت عبارتِ ریاضی ۱۴ است. عمل‌گرهای این عبارت دودویی یا یگانی هستند. کدام‌یک از گزینه‌های زیر درست است؟ (ارشد ۱۳۷۹)

۱ این عبارت بی‌گمان تعداد فردی عمل‌گر دودویی دارد.
۲ این عبارت بی‌گمان تعداد زوجی عمل‌گر یگانی دارد.
۳ این عبارت نمی‌تواند عمل‌گر دودویی داشته باشد.
۴ دستِ‌کم یک عمل‌گر یگانی در این عبارت وجود دارد.