مبانی برنامه‌سازی

نیم‌سال دوم ۹۱-۹۲

مدرس: حمید ضرابی‌زاده

دانشکده‌ی مهندسی کامپیوتر

آزمون پایان‌ترم

زمان: ۲/۵ ساعت

سؤال ۱. اعداد استرلینگ

فرض کنید $S(n,k)$ برابر تعداد راه‌های افراز یک مجموعه‌ی n عضوی به k زیرمجموعه‌ی ناتهی باشد. به طور مثال، مجموعه‌ی چهار عضوی `{1, 2, 3, 4}` را می‌توان به هفت روش زیر به دو زیرمجموعه‌ی ناتهی افراز کرد.

```
{1}, {2, 3, 4}
{2}, {1, 3, 4}
{3}, {1, 2, 4}
{4}, {1, 2, 3}
{1, 2}, {3, 4}
{1, 3}, {2, 4}
{1, 4}, {2, 3}
```

مثال بالا نشان می‌دهد که $S(۴,۲) = ۷$. اعداد $S(n,k)$، اعداد استرلینگ نوع دوم نامیده می‌شوند.

الف) اعداد استرلینگ نوع دوم را می‌توان با استفاده از فرمول زیر محاسبه کرد:

$$S(n,k) = \frac{۱}{k!}\sum_{i=۰}^{k-۱}(-۱)^i\binom{k}{i}(k-i)^n.$$

فرض کنید تابعی به نام `factorial(n)` داریم که مقدار $n!$ را محاسبه می‌کند[۱]. با استفاده از این تابع، تابعی به نام `stirling(n,k)` بنویسید که با دریافت n و k، مقدار $S(n,k)$ را بر اساس رابطه‌ی فوق به دست آورد.

ب) اعداد استرلینگ را می‌توان بر اساس رابطه‌ی بازگشتی زیر نیز تعریف نمود:

$$S(n,k) = \begin{cases} ۱ & \text{اگر } k = ۱ \text{ یا } k = n \\ S(n-۱,k-۱) + kS(n-۱,k) & \text{در غیر این صورت} \end{cases}$$

تابع `stirling` را این بار به صورت بازگشتی و بر اساس رابطه‌ی بازگشتی بالا بنویسید.

ج) همان طور که حدس می‌زنید، برنامه‌ی بازگشتی نوشته‌شده در قسمت قبل کارا نیست، زیرا این امکان وجود دارد که طی مراحل بازگشت، مقدار $S(n,k)$ به ازای مقادیر ثابت n و k چندین بار از نو محاسبه شود. با استفاده از روش به‌خاطرسپاری (memoization) و با استفاده از یک فرهنگ داده‌ای، برنامه‌ی بازگشتی نوشته‌شده در قسمت (ب) را به گونه‌ای تغییر دهید که محاسبات تکراری آن از بین برود.

[۱] این تابع در ماژول `math` تعریف شده است.

سؤال ۲. جدول سودوکو

جدول سودوکو (Sudoku) جدولی ۹ × ۹ است که هر خانه‌ی آن یک عدد صحیح از ۱ تا ۹ قرار گرفته است. یک جدول سودوکو «معتبر» است، اگر در هر سطر، هر ستون، و هر یک از نه بلوک ۳ × ۳ آن عدد تکراری وجود نداشته باشد. نمونه‌ای از یک جدول سودوکوی معتبر در زیر آمده است.

4	9	2	5	6	3	7	8	1
3	8	6	2	1	7	5	4	9
5	7	1	8	4	9	6	2	3
9	4	8	1	3	5	2	6	7
6	3	7	4	2	8	9	1	5
2	1	5	9	7	6	4	3	8
1	5	4	3	9	2	8	7	6
7	2	9	6	8	1	3	5	4
8	6	3	7	5	4	1	9	2

فرض کنید محتوای جدول سودوکو در یک لیست دو بعدی ۹ × ۹ به نام board به شما داده شده است، طوری که board[i][j] حاوی عدد موجود در سطر i و ستون j اُم است. فرض کنید که شماره‌ی سطرها و ستون‌ها از صفر شروع می‌شوند.

الف) تابعی به نام is_valid_row به شکل زیر بنویسید که با دریافت board و شماره‌ی سطر row، مشخص کند که آیا اعداد موجود در سطر row غیرتکراری هستند یا خیر. اگر اعداد غیرتکراری بودند، تابع مقدار True، و در غیر این صورت مقدار False برمی‌گرداند.

```
def is_valid_row(board, row):
```

ب) تابعی به نام is_valid_square بنویسید که با دریافت board و شماره‌ی سطر و ستون، مشخص کند که آیا اعداد موجود در مربع ۳ × ۳ که با شماره‌ی سطر و ستون داده‌شده شروع می‌شوند، غیرتکراری هستند یا خیر. اگر اعداد غیرتکراری بودند، تابع مقدار True، و در غیر این صورت مقدار False برمی‌گرداند.

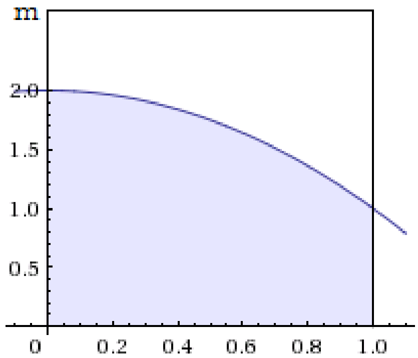
```
def is_valid_square(board, row, col):
```

ج) به کمک توابع is_valid_row و is_valid_square و همچنین تابع is_valid_column که مشابه تابع is_valid_row است و نیازی به بازنویسی آن نیست، تابعی به نام is_valid_table بنویسید که با دریافت یک جدول سودوکو مشخص کند که آیا جدول داده‌شده معتبر است یا خیر. در صورت معتبر بودن تابع مقدار True، و در غیر این صورت مقدار False برمی‌گرداند.

```
def is_valid_table(board):
```

سؤال ۳. انتگرال‌گیری

فرض کنید تابع f در بازه‌ی $[a,b]$ نامنفی و دارای کران بالای m است. می‌خواهیم انتگرال تابع f را در بازه‌ی $[a,b]$ به دست آوریم. این کار معادل با به دست آوردن مساحت محصور بین محور xها و تابع f در بازه‌ی $[a,b]$ است. به طور مثال، در شکل زیر مساحت تابع $f(x) = ۲ - x^۲$ در بازه‌ی $[۰,۱]$ نشان داده شده است.



یک روش برای تخمین مقدار این مساحت استفاده از روش مونت کارلو به صورت زیر است. ابتدا تعداد زیادی نقطه‌ی تصادفی درون مستطیل $[a,b] \times [۰,m]$ تولید می‌کنیم. سپس نسبت تعداد نقاطی که در زیر تابع f قرار می‌گیرند به کل تعداد نقاط به دست می‌آوریم. این نسبت بیان‌گر نسبت مساحت ناحیه‌ی زیر تابع به مساحت کل مستطیل است. از آن‌جایی که مساحت مستطیل مشخص است، با استفاده از نسبت به‌دست‌آمده، مساحت ناحیه‌ی زیر تابع تخمین زده می‌شود.
تابعی به نام `integral` بنویسید که با دریافت تابع f، مقادیر a و b، و کران بالای m، با تولید ۱۰۰۰ نقطه‌ی تصادفی به شکلی که در بالا توضیح داده شد، تخمینی از انتگرال f در بازه‌ی $[a,b]$ به دست آورد و مقدار به‌دست‌آمده را به عنوان خروجی تابع برگرداند.

```python
def integral(f, a, b, m):
```

راهنمایی: برای تولید یک عدد حقیقی تصادفی با توزیع یک‌نواخت در بازه‌ی $[a,b]$ می‌توانید از تابع زیر از ماژول `random` استفاده کنید.

```python
random.uniform(a, b)
```

سؤال ۴. اعداد مختلط

یک عدد مختلط در حالت کلی به صورت $a+bj$ نشان داده می‌شود که در آن a و b اعدادی حقیقی و j نشان‌دهنده‌ی $\sqrt{-1}$ است. به طور مثال $2 + 3j$ یک عدد مختلط است. در این سوال قرار است کلاسی برای نگه‌داری اعداد مختلط و کار با آن‌ها طراحی کنیم.
کلیات این کلاس در زیر نشان داده شده است.

```
class complex:
    ''' Represent complex numbers '''

    def __init__(self, a, b):
        self.a = a
        self.b = b

    def __str__(self):

    def add(self, c):

    def mult(self, c):
```

این کلاس دارای توابع زیر است:

- تابع `__init__` که هنگام ایجاد هر نمونه (عدد مختلط)، آن را مقداردهی اولیه می‌کند.
- تابع `__str__` که عدد مختلط جاری (نگه‌داری‌شده در `self`) را در قالب یک رشته به شکل $(a + bj)$ برمی‌گرداند.
- تابع `add` که عدد مختلط جاری را با عدد مختلط c جمع می‌کند و نتیجه را در قالب یک عدد مختلط جدید برمی‌گرداند.
- تابع `mult` که عدد مختلط جاری را با عدد مختلط c ضرب می‌کند و نتیجه را در قالب یک عدد مختلط جدید برمی‌گرداند.

تابع `__init__` به طور کامل برای شما نوشته شده است. سه تابع دیگر را مطابق تعاریف بالا کامل کنید.

نمونه‌ای از نحوه‌ی کاربرد این کلاس در زیر آورده شده است:

```
>>> x = complex(1, 2)
>>> y = complex(2, 3)
>>> print(x, y)
(1 + 2j) (2 + 3j)
>>> print(x.add(y))
(3 + 5j)
>>> print(x.mult(y))
(-4 + 7j)
```