# Республиканская студенческая олимпиада по специальности «Математическое и компьютерное моделирование» Астана, 01.04.16 Решения

1. Введём функцию

$$f(n) = \left[\sqrt{1}\,\right] + \left[\sqrt{2}\,\right] + \left[\sqrt{3}\,\right] + \ldots + \left[\sqrt{n^2-1}\,\right] + \left[\sqrt{n^2}\,\right],$$

где $[x]$ — наибольшее целое число, не превышающее $x$. Напишите **код функции**, которая вычисляет $f(n)$ для данного натурального $n$, не используя при этом операцию извлечения корня и вещественную арифметику.

*Баллы начисляются в зависимости от асимптотической сложности алгоритма.*

*(Абдикалыков А.К.)*

*Решение:*

Максимальный балл давался за алгоритм с асимптотической сложностью $O(1)$ (явную формулу), промежуточные баллы — за сложность $O(n)$ и $O(n^2)$.

$$\sum_{j=1}^{n^2}\left[\sqrt{j}\,\right] = \sum_{k=1}^{n}\left(k\cdot\sum_{\substack{[\sqrt{j}\,]=k\\ 1\leqslant j\leqslant n^2}}1\right) = \sum_{k=1}^{n-1}\left(k\cdot\sum_{\substack{[\sqrt{j}\,]=k\\ 1\leqslant j\leqslant n^2}}1\right) + n =$$

$$= \sum_{k=1}^{n-1}\left(k\cdot\sum_{j=k^2}^{(k+1)^2-1}1\right) + n = \sum_{k=1}^{n-1}k(2k+1) + n = \sum_{k=1}^{n-1}(2k^2+k) + n =$$

$$= 2\cdot\frac{(n-1)n(2n-1)}{6} + \frac{(n-1)n}{2} + n = \frac{(n-1)n(4n+1)}{6} + n = \frac{n(4n^2-3n+5)}{6}$$

```
function f(n)
    return n*(4*n*n-3*n+5)/6
```

2. На декартовой координатной плоскости нарисованы две полупараболы: график функции $y = x^2$ $(x \geqslant 0)$ и его копия, повёрнутая на прямой угол по часовой стрелке. Эти две кривые отсекают от прямой, параллельной оси ординат, отрезок длины $L$. Обозначим через $S(L)$ — площадь отсечённой фигуры.

а) Докажите, что $S(L) > 1$ при $L > 2$;

б) Напишите **код функции**, которая вычисляет $S(L)$ для данного положительного вещественного числа $L$.

*(Абдикалыков А.К.)*

*Решение:*

а) Две полупараболы из условия задачи — это графики функций $y = x^2$ и $y = -\sqrt{x}$ при $x \geqslant 0$. Поэтому искомая функция

$$S(L) = \int\limits_0^{f^{-1}(L)} f(x)\,dx,$$

где $f(x) = x^2 + \sqrt{x}$. (В силу монотонности функции $f(x)$ корректно вводить $f^{-1}(L)$.) Так как, кроме прочего, подынтегральная функция в определении $S(L)$ положительная, то и сама функция $S(L)$ — возрастающая. Поскольку $S(2) = 1$ (это можно показать разными способами: как графически, составив квадрат, например, так и аналитически, посчитав явно интеграл), то $S(L) > 1$ при $L > 2$.

б) Найдём сначала $x_0 = f^{-1}(L)$ с помощью бинарного поиска, затем вычислим

$$S(L) = \left(\frac{1}{3}x^3 + \frac{2}{3}x^{\frac{3}{2}}\right)\Bigg|_{x=0}^{x_0} = \frac{x_0^3 + 2x_0^{\frac{3}{2}}}{3} = \frac{x_0(2x_0^2 + 2\sqrt{x_0} - x_0^2)}{3} = \frac{x_0(2L - x_0^2)}{3}.$$

```
function S(L)
    left = 0
    right = sqrt(L)
    while right - left > eps
        med = (left + right) / 2
        if med * med + sqrt(med) > L
            right = med
        else
            left = med
    return med * (2 * L - med * med) / 3
```

3. Найти все дифференцируемые функции $f\colon \mathbb{R} \to \mathbb{R}$, удовлетворяющие соотношению

$$f(x-y) + f(x+y) = f'(x^2 + y^2)$$

для любых $x, y \in \mathbb{R}$.

*(Баев А.Ж.)*

*Решение:*

Продифференцируем по $x$ и $y$:

$$f'(x-y) + f'(x+y) = 2xf''(x^2 + y^2),$$
$$-f'(x-y) + f'(x+y) = 2yf''(x^2 + y^2).$$

Пусть $x \neq 0$, $y \neq 0$. Приравняем $f''(x^2 + y^2)$:

$$(y + x)f'(x-y) = (x-y)f'(x+y).$$

Пусть $|x| \neq |y|$.

$$\frac{f'(x+y)}{x+y} = \frac{f'(x-y)}{x-y}.$$

Зафиксируем величину $x - y = A$, отличную от нуля. Тогда выражение справа не зависит от $y$ и равно некоторой константе $2C$.

$$\frac{f'(2y+A)}{2y+A} = \frac{f'(A)}{A} = 2C.$$

Заметим, что $t = 2y + A$ может принимать любые ненулевые значения. Значит, при $t \neq 0$:

$$f'(t) = 2Ct.$$
$$f(t) = Ct^2 + C_1.$$

После подстановки в исходное уравнение, получим: $C_1 = 0$. При $t = 0$ доопределяется из непрерывности $f'(t)$ (по соотношению в условии). Ответ: $f(t) = Ct^2$.

4. Дана функция $f\colon [0,2n] \to \mathbb{R}$. Пусть $f_i = f(i)$ — значения функции во всех целых $i$ от 0 до $2n$. Дана переменная $S$ вещественного типа с начальным значением 0. За один ход робот может выбрать целое $i$ от 1 до $2n-1$, затем добавить к переменной $S$ или вычесть из нее среднее арифметическое значений функции $f(x)$ в узлах $i-1$, $i$, $i+1$:

$$S := S \pm \frac{f_{i-1} + f_i + f_{i+1}}{3}.$$

Может ли робот за конечное число ходов получить в переменной $S$ значение

$$I = \frac{1}{3}\left(f_0 + 4\sum_{k=1}^{n} f_{2k-1} + 2\sum_{k=1}^{n-1} f_{2k} + f_{2n}\right),$$

которое является приближением интеграла $\int\limits_0^{2n} f(x)\,dx$, если

а) $f(0) = f(2n) = 0$;
б) $f(0) \neq 0$, $f(2n) \neq 0$?

*(Баев А.Ж., Абдикалыков А.К.)*

*Решение:*

Пусть конечное значение $S = \sum\limits_{j=1}^{2n-1} c_j \cdot \frac{f_{j-1}+f_j+f_{j+1}}{3}$. Тогда пункт б) эквивалентен решению нижеуказанной системы линейных уравнений, причём в целых числах. Видно, что система состоит из $(2n+1)$ уравнения относительно $(2n-1)$ неизвестной.

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \cdots \\ c_{2n-3} \\ c_{2n-2} \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 2 \\ 4 \\ \cdots \\ 4 \\ 2 \\ 4 \\ 1 \end{pmatrix}$$

Рассматривая только первые $(2n-1)$ уравнения, мы получим систему с нижнетреугольной матрицей и определителем, равным единице, а значит, всеми уравнениями, кроме последних двух, все неизвестные определяются однозначно, принимая при этом целые значения. Таким образом, задача сводится к нахождению таких $n$, чтобы система из этих $(2n-1)$ уравнения имела решение, совместимое с дополнительными условиями $c_{2n-2} + c_{2n-1} = 4$, $c_{2n-1} = 1$. Решая эту систему методом Гаусса, получаем

$$\begin{gathered} c_1 = 1, \quad c_2 = 3, \quad c_3 = -2, \quad c_4 = 3, \quad c_5 = 1, \quad c_6 = 0, \\ c_7 = 1, \quad c_8 = 3, \quad c_9 = -2, \quad c_{10} = 3, \quad c_{11} = 1, \quad c_{12} = 0, \\ \cdots \end{gathered}$$

Таким образом, равенства $c_{2n-2} = 3$, $c_{2n-1} = 1$ выполняются только в том случае, если

$$2n - 1 = 5 \pmod 6,$$

или, что то же самое, $n$ кратно 3.

Пункт а) эквивалентен решению той же системы в целых числах, но уже без первого и последнего уравнений.

$$\begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \cdots \\ c_{2n-3} \\ c_{2n-2} \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 4 \\ \cdots \\ 4 \\ 2 \\ 4 \end{pmatrix}$$

Фиксируя $c_1 = c$ и используя все уравнения, кроме последнего ($c_{2n-2} + c_{2n-1} = 4$), находим

$$\begin{array}{cccccc} c_1 = c, & c_2 = 4 - c, & c_3 = -2, & c_4 = 2 + c, & c_5 = 2 - c, & c_6 = 0, \\ c_7 = c, & c_8 = 4 - c, & c_9 = -2, & c_{10} = 2 + c, & c_{11} = 2 - c, & c_{12} = 0, \end{array}$$
$$\cdots$$

Значит,

$$c_{2n-2} + c_{2n-1} = \begin{cases} c, & 2n - 2 = 0 \pmod 6, \\ -2 - c, & 2n - 2 = 2 \pmod 6, \\ 4, & 2n - 2 = 4 \pmod 6. \end{cases}$$

Видно, что в любом случае можно подобрать такое $c$, чтобы выполнялось равенство

$$c_{2n-2} + c_{2n-1} = 4,$$

из чего следует, что система совместна при любом $n$.

5. Дана некоторая условная машина, состоящая из памяти в $n$ бит и указателя, который в каждый отдельный момент находится над какой-то из этих n ячеек. Перед запуском программы в память записывается некоторое натуральное число $m$ в двоичной системе счисления, а указатель устанавливается над крайним правым (младшим) битом числа. Язык программирования для этой машины состоит из следующих команд:

| | | |
|---|---|---|
| **L** | left | сместить указатель налево на одну ячейку, если это возможно, иначе завершить программу |
| **R** | right | сместить указатель направо на одну ячейку, если это возможно, иначе завершить программу |
| **C** | change | изменить значение бита в текущей ячейке на противоположное |
| **A** | again | перейти к выполнению первой команды |
| **S** | skip | пропустить две следующие команды, если в текущей ячейке 0 |
| **F** | finish | завершить выполнение программы |

Команды записываются в одну строку и выполняются в последовательном порядке, слева направо. При этом запись программы обязана оканчиваться командой **A** или **F**. Напишите для этой абстрактной машины следующие программы:

а) заменить данное число на $(m - 1)$;

б) заменить данное число на $(2^n - m - 1)$;

в) изменить на противоположный его старший (крайний слева) бит.

*Примеры:*

а) программа, обнуляющая все ячейки: **SSCLA**;

б) программа, которая изменяет второй справа бит, если крайний справа бит нулевой: **SFFLCF**.
*(Абдикалыков А.К.)*

*Решение*

а) Уменьшить двоичное число на единицу: **CSLAF**.

б) Поменять все биты: **CLA**.

в) Поменять только старший бит: **CLRCLA**.

6. Из квадратной однородной пластины со стороной 1 случайным образом вырезается квадрат со сторонами, равными $2a$ и параллельными сторонам исходного квадрата. При этом центр квадрата — это случайная величина, равномерно распределённая по всем допустимым положениям (квадрат со стороной $(1-2a)$).

   а) Найдите вероятность $p(a)$ того, что центр тяжести полученной фигуры лежит в вырезанной области.

   б) Напишите **код функции** $p(a)$, которая вычисляет указанную вероятность приблизительно, считая при этом, что нам не известен метод нахождения центра тяжести произвольной фигуры, однако мы можем найти центр тяжести конечного множества точек одинаковой массы.

   *(Баев А.Ж.)*

*Решение*

а) Пусть исходный квадрат — это квадрат $[0,1]\times[0,1]$ на плоскости, а центр вырезанного квадрата — $(x_0, y_0)$. Чтобы квадрат целиком поместился, то $(x_0; y_0) \in [a, 1-a]\times[a, 1-a]$.

1 шаг. Найдем центр тяжести. Запишем функцию плотности массы пластины по оси $OX$:

$$f(x) = \begin{cases} 1, x < x_0 - a \\ 1-2a, x \in [x_0 - a, x_0 + a] \\ 1, x < x_0 + a \end{cases}$$

Найдем проекцию центра тяжести $m$ на ось $OX$:

$$m = \frac{\int_0^1 xf(x)dx}{\int_0^1 f(x)dx} = \frac{1-8a^2x_0}{2(1-4a^2)}.$$

2 шаг. Найдем вероятность попадания центра тяжести в вырезанную часть.

$$P = P(m \in [x_0 - a; x_0 + a]) = P\left(\frac{1-8a^2x_0}{2(1-4a^2)} < x_0 + a\right) - P\left(\frac{1-8a^2x_0}{2(1-4a^2)} < x_0 - a\right) =$$

$$= P\left(x_0 + a > \frac{1}{2} + 4a^3\right) - P\left(x_0 - a > \frac{1}{2} - 4a^3\right) = P_1 - P_2.$$

Вычислим $P_1$. Заметим, что $x_0 + a$ равномерно распределено на $[2a, 1]$. Поэтому важно понять, попадает ли $\frac{1}{2} + 4a^3$ в интервал $[2a, 1]$. $\frac{1}{2} + 4a^3 < 1$ ввиду того, что $a \in [0, \frac{1}{2}]$. Проверим левую границу:

$$\frac{1}{2} + 4a^3 > 2a$$

$$\left(a - \frac{1}{2}\right)(a-\varphi)(a-\overline{\varphi}) > 0$$

где $\varphi = \frac{-1+\sqrt{5}}{4}$. Значит, $\frac{1}{2} + 4a^3$ попадает в интервал $[2a, 1]$ при $a < \varphi$.

$$P_1 = \begin{cases} \frac{1-8a^3}{2(1-2a)}, & a < \varphi \\ 1, & a > \varphi. \end{cases}$$

Аналогично вычислим $P_2$. $x_0 - a$ равномерно распределено на $[0, 1-2a]$. Поэтому важно понять, попадает ли $\frac{1}{2} - 4a^3$ в интервал $[0, 1-2a]$. $\frac{1}{2} - 4a^3 > 0$ ввиду того, что $a \in [0, \frac{1}{2}]$. Значит:

$$P_2 = \begin{cases} \frac{1-4a+8a^3}{2(1-2a)}, & a < \varphi \\ 0, & a > \varphi. \end{cases}$$

Найдем вероятность попадания центра тяжести в вырезанную часть:

$$(P_1 - P_2)^2 = \begin{cases} 4a^2(1+2a)^2, & a \in [0, \frac{-1+\sqrt{5}}{4}] \\ 1, & a \in \left[\frac{-1+\sqrt{5}}{4}, \frac{1}{2}\right] \end{cases}$$

б) Промоделируем методом Монте–Карло и подсчет центра тяжести, и подсчет ответа. Генерируем $N$ подходящих квадратов. У каждого из них генерируем $M$ случайных точек. Если центр тяжести данных точек находится внутри квадрата, то засчитываем этот квадрат. Иначе — нет. Отметим, что порядок аппроксимации данного метода $O\left(\frac{1}{\sqrt{NM}}\right)$.

```
function p(a)
    N = 1000
    M = 1000
    ans = 0
    for i = 1 .. N
        x0 = rand(a, 1 - a)
        y0 = rand(a, 1 - a)
        m = 0
        mx = 0
        my = 0
        for j = 1 .. M
             x = rand(0, 1)
             y = rand(0, 1)
             if not ((x, y) in [x0 - a, x0 + a]*[y0 - a, y0 + a] then
                 mx = mx + x
                 my = my + y
                 m = m + 1
        mx = mx / m
        my = my / m
        if (mx, my) in [x0 - a, x0 + a]*[y0 - a, y0 + a] then
            ans = ans + 1
    return ans / N
```