Ответы, указания и критерии проверки

Описание критериев:

Критерии.

+1 Означает, что описанная в пункте часть решения стоит 1 балл.

-1 Означает, что за описанную в пункте ошибку снимается 1 балл.

2 По-умолчанию означает максимальный балл (2) за описанный случай. Иные трактовки поясняются.

Задачи

Часть I

Задача 1(2). Верно ли, что $f(n) = \Theta(g(n))$ тогда и только тогда, когда $\exists C > 0 : \lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = C$?

Решение. Нет, неверно. Рассмотрим колеблющуюся функцию: $f(n) = 2 + \sin n$, к примеру; $g(n) = 1$. Тогда $1 \leqslant 2 + \sin n \leqslant 3$, $f(n) = \Theta(g(n))$. При этом $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \lim\limits_{n\to\infty} \frac{2+\sin n}{1} = \lim\limits_{n\to\infty} (2 + \sin n)$, а этот предел не существует (и значит, не может быть равен никакой константе).

Критерии.

0 Ответ «верно».

0,5 Утверждение, что если константы оценок сверху и снизу различны, то предела может не быть (без доказательства и контрпримера).

1 Идейно верный пример (колеблющаяся функция), есть отрицательные числа среди значений.

Задача 2(3). На вход подаются числа n и k и массив целых чисел $a_1, \ldots, a_n$. Определим частичную сумму $S_m = \sum\limits_{i=1}^{m} a_i$. Необходимо вывести значение k-й по величине частичной суммы (у i-й и $(i+1)$-й по величине суммы могут совпадать значения).

Решение. Составляется массив частичных сумм ($S_1 := a_1,\ S_k := S_{k-1} + a_k$), затем в нём ищется k-я порядковая статистика. Оба этапа совершаются за линейное время, поэтому суммарное время тоже линейное.

Критерии.

- 0.5 балла за решение, если алгоритм некорректен, однако в результате его работы считается массив частичных сумм (например, если решение состоит в том, что алгоритм находит просто k-ю частичную сумму, а не k-ю в порядке возрастания).

- 1 балл за решение,если описан корректный алгоритм, который работает за $\Theta(n\ logn)$.
- -1 балл, если неправильно/неэффективно указана асимптотика использованного алгоритма (например, "найдём k-ю порядковую статистику за $O(n\ logn)$".

Задача 3 (3). Вычислить быстрое преобразование Фурье вектора $[1, 3, 1, 2]$.

Критерии.

0	ошибки в первом или втором вычисление (хоть из решения и примерно ясен алгоритм)
0	преобразование (расширенного) вектора длины 8 посчитано не до конца или с ошибкой
0,5	нет промежуточных вычислений / нет «быстрого» алгоритма
1	студент примерно знает алгоритм, но не поддалось последнее вычисление
2	верно вычислено преобразование (расширенного) вектора длины 8
2	перепутан порядок корней
2,5	алгоритм «быстрый», но явно делаются лишние вычисления

Задача 4 (3). Открытый ключ в протоколе RSA $(61, 437)$. Известно, что
– $437 = 19 \cdot 23$, 19 и 23 – простые;
– обратное к числу 61 в кольце вычетов по модулю 437 есть 43;
– обратное к числу 61 в кольце вычетов по модулю 396 есть 13.
Вы хотите отправить сообщение 13 и подписать его своей электронной подписью.
Также известно, что
– $13^{12} = 197$ в кольце вычетов по модулю 437;
– $13^{42} = 49$ в кольце вычетов по модулю 437;
– $13^{12} = 37$ в кольце вычетов по модулю 396;
– $13^{42} = 37$ в кольце вычетов по модулю 396.
Чему равна последняя цифра подписанного сообщения?

Решение. Секретная компонента $d = e^{-1}$ в кольце вычетов по модулю $\phi(n) = 396$. Подпись $s = m^d \mod n$ Ответ: 6. Сама подпись 376.

Критерии.

- 0 за ошибку в алгоритме (вычисления не в том кольце, возведение не в ту степень и т.д.)
- от -0.5 до -1 за каждую арифметическую ошибку

Задача 5 (3). Найдите асимптотику роста функций, полагая, что $T(n) = \Theta(1)$ при малых n:

а) $T(n) = 27\,T\left(\frac{n}{3}\right) + n^3$; **б)** $T(n) = 256\,T\left(\frac{n}{16}\right) + 2n \times \frac{\sqrt{n} + \log n}{1 - \frac{1}{n}}$.

Решение.
а) $d = \log_3 27 = 3 \Rightarrow$ по 2 пункту основной Теоремы ответ $\Theta(n^3 \cdot \log n)$.

б) Несложно заметить, что начиная с $N = 2$ и для констант $c_1 = 2, c_2 = 8$ выполнено, что $2n\sqrt{n} \leqslant 2n \times \frac{\sqrt{n}+\log n}{1-\frac{1}{n}} \leqslant 8n\sqrt{n} \Rightarrow 2n \times \frac{\sqrt{n}+\log n}{1-\frac{1}{n}} = \Theta(n^{1.5})$. При этом $d = \log_{16} 256 = 2 \Rightarrow$ по 1 пункту основной Теоремы, взяв $\varepsilon = 0.1$ и проверяя, что $2n \times \frac{\sqrt{n}+\log n}{1-\frac{1}{n}} = O(n^{d-\varepsilon})$, получаем ответ $\Theta(n^2)$.

Критерии.

- По 1.5 балла за каждый пункт
- За первый пункт ставится 1.5 балла в случае правильного ответа с обоснованием (ссылкой на Теорему), иначе – 0.
- За второй пункт ставится 1.5 балла в в случае правильного ответа с обоснованием (ссылкой на Теорему).

  Снимается 0.5 баллов в случае, когда допущена несерьёзная арифметическая ошибка;

  0.5 баллов за то, что не указали конкретный ε;

  1 балл за серьёзную ошибку, которая потенциально могла повлиять на ответ (в основном когда перепутали числитель и знаменатель при раскрытии $\frac{1}{n}$)
- При решении не через Теорему решение оценивается индвидиуально, исходя из обоснованности ответа и правильности рассуждений.

Задача 6(3). Игральный кубик с тремя сторонами (-1, 0 и 1) бросают четыре раза и получают матрицу 2×2. Найдите вероятность того, что будет получена матрица с детерминантом, равным 1.

Решение. Назовем числа, выпавшие на кубике на каждой попытке соответственно: a, b, c, d. Запишем их в матрицу и запишем ее детерминант: $\det M = ad - bc$. Нам нужно, чтобы получилась матрица с детерминантом 1.

Произведения ad, bc могут принимать значения $\{-1, 0, 1\}$. Заметим, что нам подойдут только те случаи, когда $A = \{ad = 1, bc = 0\}$ или $B = \{ad = 0, bc = -1\}$. В других случаях детерминант матрицы будет отличным от 1.

Для события A подходящими парами (a,d) будут: $(-1, -1), (1, 1)$, для (b, c) – (0,0), (0,1), (1,0), (-1,0), (0,-1). Пары $(a, d), (b, c)$ – независимые, количество вариантов для каждой из таких пар – 3^2

Тогда $\mathbb{P}\{A\} = \frac{2}{3^2} \cdot \frac{5}{3^2} = \frac{10}{81}$

Рассмотрим событие B. $(a, d) \in \{(0, 0), (0, 1), (1, 0), (-1, 0), (0, -1)\}$, $(b, c) \in \{(1, -1), (-1, 1)\}$

Тогда $\mathbb{P}\{B\} = \frac{5}{3^2} \cdot \frac{2}{3^2} = \frac{10}{81}$

События A, B имеют нулевое пересечение, значит $\mathbb{P}\{A \cup B\} = \frac{20}{81}$

Критерии.

0 Неверный ответ, либо верный ответ без каких бы то ни было обоснований.

0.5 Приведена верная формула для рассчета итоговой вероятности в общем виде, проведены некоторые подсчеты, но в них допущена смысловая неарифметическая ошибка.

2.5 Решение верное, но была допущена **арифметическая** ошибка.

3 Правильный ответ с корректным обоснованием.

Часть II

Задача 7 (4). В этой задаче битовая модель вычислений.

1. Оцените асимптотически время работы следующего алгоритма. На вход алгоритма поступает массив координат n точек на плоскости (x_i, y_i), где x_i, y_i — n-битовые числа; алгоритм сортирует эти точки по расстоянию от нуля следующим образом. Для каждой точки производится вычисление суммы квадратов координат, возведение в квадрат вычисляется алгоритмом Карацубы. После этого выполняется сортировка слиянием.

2. Верно ли, что $T(n) = O(n^2)$, где $T(n)$ — времени работы алгоритма?

Решение.

1.
   - Одна пара координат - два числа длины n. Возведение их в квадрат алгоритмом карацубы - $O(n^{\log_2 3})$. Последующее сложение в силу того что длина квадратов не превышает $2n$ линейно, т.е. общая асимптотика возведения в квадрат и сложения занимает $O(n^{\log_2 3})$. Для n пар координат соответственно $O(n^{1+\log_2 3})$.
   - Сортировка слиянием в атомарной модели вычислений имеет асимптотику $O(n \log n)$. Учитывая асимптотику одного сравнения $O(n)$ получаем, что итоговая асимптотика сортировки слиянием в битовой модели вычислений - $O(n^2 \log n)$.
   - Итоговая сложность $O(n^{1+\log_2 3}) + O(n^2 \log n) = O(n^{1+\log_2 3})$

2. Да, т.к. по определению $T(m)$ - это функция от длины входа, который представляет из себя n пар чисел длины n, т.е. $m = n^2$. Таким образом, $n^{1+\log_2 3} \approx n^{2.58} = O(n^4)$, поэтому $T(m) = O(m^2)$.

Критерии.

+1 Корректно приведена асимптотика алгоритма Карацубы и общая асимптотика подсчета расстояния до нуля для всех точек.

+0.5 Корректно указана асимптотика сортировки слиянием $O(n \log n)$ без переложения на битовую модель вычислений.

+1.5 Корректно указана асимптотика сортировки слиянием $O(n \log n)$, предпринята попытка получить асимптотику для битовой модели вычислений, не приведшая к положительному исходу (например построение рекурренты $T(n) = 2T(\frac{n}{2}) + n^2$)

+3 Корректно указана асимптотика сортировки слиянием в битовой модели вычислений - $O(n^2 \log n)$, в результате чего получена корректная асимптотика всего алгоритма и ответ на второй пункт.

Задача 8(3). В оперативой памяти хранится k массивов $A^1, A^2, \ldots, A^k$, в каждом из которых хранятся числа в диапазоне от 1 до n, и сумма длин массивов (общее число элементов) также равна n. Постройте эффективный алгоритм, сортирующий все массивы.

Решение.

1. Для каждого элемента A^i_j запишем пару (A^i_j, i) (само значение и номер массива из которого оно пришло), получим массив длины n таких пар (по условию всего чисел в массиве – n). Сложность $O(n)$

2. Отсортируем этот массив пар с помощью сортировки подсчетом по первой компоненте. Первая компонента может принимать целые значения из $[1, n]$, всего элементов – n, таким образом сложность будет $O(n + n) = O(n)$

3. Теперь пройдем по отсортированному массиву пар, и для пары (x, y) будем записывать x в y-й массив ответа. Сложность $O(n)$

Итоговая сложность: $O(n)$ по времени/дополнительной памяти.

Доказательство корректности. Рассмотрим массив A^i. В отсортированном массиве пар элементы, что пришли из него, будут лежать в порядке возрастания, а во второй компоненте у них будет записано i – номер массива из которого они пришли. Таким образом в третьем пункте мы их запишем в нужный массив в отсортированном порядке.

Критерии.

3 Корректное решение за $O(n)$

0.5 Корректное решение за $O(n \log n)$ (к примеру сделан массив пар, но затем применена сортировка слиянием), или $O(kn)$ (k раз применена сортировка подсчетом)

0 Корректное решение, но медленнее чем $O(n \log n)/O(kn)$

0 Некорректное решение, решение не той задачи (в частности решения где массив просто сливался из k отдельных в один, сортировался, и это выдавалось в качестве ответа)

0 Попытка использовать сортировку подсчетом без учета сложности на хранение диапазона чисел (обычно формулировка вида "Сложность O(n) т.к. $\sum_i O(|A^i|) = O(n)$")

-0.5 – -1.0 Решение корректно, но есть мелкие недочеты, ошибки при подсчете сложности и т.д.

Задача 9(3+4). В оперативной памяти хранится массив целых чисел $a_1, a_2, \ldots, a_n$. Постройте алгоритм, который проверяет, есть ли такое число $i \in \{1, \ldots, n\}$, что $a[i] = i$ и выводит любое i в случае положительного ответа, если известно, что **а)** a — (строго) возрастающий массив;

б) a — неубывающий массив. Докажите оптимальность своего алгоритма (в каждом пункте).

Указания. а) оптимальным является бинарный поиск (доказательство аналогично оценке для прообраза монотонной функции); **б)** оптимальным является просмотр всего массива; стратегия противника: до последнего шага на запрос $a[i]$ возвращать $i - 1$.

Критерии.

0.5 Корректный алгоритм для пункта а)

0.5 Корректность, сложность описанного алгоритма для пункта а)

2 Доказательство оптимальности для пункта а)

1 Алгоритм + корректность + сложность для пункта б)

3 Доказательство отптимальности для пункта б)

Задача 10 (5). Рассмотрим функцию $s(n)$, определенную следующим образом:

$s(1) = 1$

$s(2) = 1, 2, 1$

$s(3) = 1, 2, 1, 3, 1, 2, 1$

$s(n+1) = s(n), n+1, s(n)$

Найти математическое ожидание количества совпадений $u[i] = v[i]$ у двух случайных циклических сдвигов u и v массива $s(n)$. Циклическим сдвигом массива $a = a_1 a_2 \dots a_n$ называется массив вида $a_i a_{i+1} \dots a_n a_1 \dots a_{i-1}$. Ответ должен быть представлен в виде алгебраического выражения, содержащего суммы константного числа слагаемых.

Указания. Количество вхождений числа k в последовательность $s(n)$ равно 2^{n-k}. Длина последовательности $s(n)$ равна $2^n - 1$. Оба утверждения доказываются по индукции. Найдём сначала условную вероятность совпадения одного элемента последовательностей u и v в зафиксированной позиции, при условии, что в u на этой позиции стоит символ k:

$$\Pr\left[u[i] = v[i] \,\middle|\, u[i] = k\right] = \Pr\left[v[i] = k\right] = \Pr\left[u[i] = k\right] = \frac{2^{n-k}}{2^n - 1}$$

Воспользуемся формулой полной вероятности:

$$\Pr\left[u[i] = v[i]\right] = \sum_{k=1}^{n} \Pr\left[u[i] = v[i] \,\middle|\, u[i] = k\right] \times \Pr\left[u[i] = k\right] = \sum_{k=1}^{n} \left(\frac{2^{n-k}}{2^n - 1}\right)^2 = \frac{4^n - 1}{3(2^n - 1)^2}$$

Для вычисления искомого математического ожидания случайной величины f введём индикаторные случайные величины g_i, возвращающие 1 при совпадении i-х элементов последовательностей. Тогда

$$\mathrm{E}\big[f\big] = \mathrm{E}\left[\sum_{i=1}^{2^n-1} g_i\right] = \sum_{i=1}^{2^n-1} \mathrm{E}\big[g_i\big] = (2^n - 1) \times \Pr\big[u[i] = v[i]\big] = \frac{4^n - 1}{3(2^n - 1)}$$

Критерии.

1 Найдены границы возможного числа совпадений или найдено число вхождений каждого числа.

-1 Не обосновано сведение вычисления матожидания к нахождению среднего числа совпадений (в случае решения через явный подсчет по всевозможным сдвигам).

4 Получена корректная сумма.

5 Полностью правильное решение и правильный ответ, представленный в форме, содержащей суммы константного числа слагаемых.

Задача 11 (6). В оперативной памяти хранится массив из целых чисел $a_1, a_2, \dots, a_n$. Нужно разрезать его на три части $a_1 \dots a_i$, $a_{i+1} \dots a_j$ и $a_{j+1} \dots a_n$ так, чтобы суммы элементов хотя бы двух частей из трёх были положительны (гарантированно, что такое разбиение существует). Постройте эффективный алгоритм, решающий задачу (выход: индексы i и j).

Решение.

1. Как искать решение, если положительные две крайние части: идти от начала, считая частичную сумму, пока не встретим первую положительную, если такая есть. То же самое делаем с конца, если полученные префикс и суффикс не пересеклись, то решение найдено.

2. Поиск решения, если положительные 2 и 3 части аналогичен поиску решения для 1 и 2 части

3. Как искать решение, если положительные первые две части: Посчитаем частичные суммы $s_k = \sum_{i=1}^{k} a_i$. Заметим, что положительность первых двух частей эквивалентна тому, что $0 < s_i < s_j$. Пройдемся по массиву частичных сумм, рассматривая только положительные, если эта последовательность из положительных монотонно невозрастает, то такой пары i, j нет. Иначе найдутся две положительные суммы, такие что $0 < s_i < s_j$, которые задают искомое разбиение

Критерии.

5 $O(n \log n)$ решение

1 $O(n^2)$ решение

0 решение медленнее $O(n^2)$

0 разбор случая, когда положительные куски расположены по бокам не приносит дополнительных баллов