

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Косенок Сергей Михайлович  
Должность: ректор  
Дата подписания: 15.06.2026 11:08:20  
Уникальный программный ключ:  
e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

**Тестовое задание для диагностического тестирования по дисциплине:**

**Алгоритмы и структуры данных**

|                          |   |
|--------------------------|---|
| Квалификация выпускника  | бакалавр  |
| Направление подготовки   | 01.03.02<br>«Прикладная математика и информатика» |
| Направленность (профиль) | «Технологии программирования и анализ данных»     |
| Форма обучения           | Очная   |
| Кафедра-разработчик      | Прикладной математики                             |
| Выпускающая кафедра      | Прикладной математики                             |

**Диагностический тест по дисциплине «Алгоритмы и структуры данных» за третий семестр**

| Проверяемые компетенции  | Задание   | Варианты ответов   | Тип сложности |
|--|---|--|---------------|
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 1) Рекуррентное выражение временной сложности алгоритма $T(N)$ определяется выражением $T(N) = T(N-1) + 1$ , если $N > 1$ и $T(N) = 1$ в противном случае.<br>Асимптотическая сложность алгоритма равна | 1) $N$<br>2) $N^2$<br>3) $N^N$<br>4) $N^3$   | низкий        |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 2) Рекуррентное выражение временной сложности алгоритма $T(N)$ определяется выражением $T(N) = 2T(N/2) + N$ , если $N > 1$ , $T(1) = 0$ . Асимптотическая сложность алгоритма равна                     | 1) $\ln N$<br>2) $N \ln N$<br>3) $N^2 \ln N$<br>4) $N^2$   | низкий        |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 3) Модификация сортировкой вставками сортировки слиянием позволяет  | 1) Получить естественную сортировку<br>2) Улучшить временные характеристики сортировки<br>3) Уменьшить требованиям по памяти<br>4) Уменьшить асимптотическую сложность от $N^2$ до $N \log(N)$ | низкий        |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1                       | 4) Алгоритм сортировки распределяющим подсчетом не используют для сортировки строк потому, что  | 1) Он не обладает необходимыми временными характеристиками<br>2) Требуется дополнительной памяти<br>3) Применим к целым числам<br>4) Имеет линейную асимптотическую сложность                  | низкий        |

|  |  |  |         |
|--|--|--|---------|
| ОПК-5.2<br>ОПК-5.3   |  |  |         |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 5) Алгоритм последовательного поиска в худшем случае при неудачном поиске имеет асимптотическую сложность  | 1) $O(1)$<br>2) $O(N)$<br>3) $O(\log N)$<br>4) Нет правильных вариантов ответов                    | низкий  |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 6) Временная сложность некоторого алгоритма определяется выражением $f(N)=N^3/3 + (10N \cdot \ln N)^2$ . Асимптотическая сложность $O(f(N))$ будет равна (выберите два подходящие варианта ответов)                                    | 1) $N^3/3$<br>2) $N^3$<br>3) $(10N \cdot \ln N)^2$<br>4) $(N \cdot \ln N)^2$                       | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 7) Какая структура данных обеспечивает эффективность добавление в начало, имеющую сложность $O(1)$ (выберите три подходящие варианта ответов)  | 1) связный список<br>2) стек<br>3) очередь<br>4) дерево  | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 8) Пусть алгоритм представлен следующим псевдокодом:<br><pre> for (i = 0; i &lt; N/2; i++) {     for (j = 0; j &lt; N/3; j++) {         f(N, other);     } } </pre> причем асимптотическая сложность $f(N, other)$ составляет $O(N)$ . | 1) $O(N^2)$<br>2) $O(N)$<br>3) $O(N \log N)$<br>4) $O(N^2 \log N)$<br>5) $O(N^3)$<br>6) $O(N^3/6)$ | средний |

|  |   |   |         |
|--|---|---|---------|
|  | Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов)  |   |         |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 9) Пусть алгоритм представлен следующим псевдокодом:<br><pre>for (i = N; i &gt; 0; i /= 2) {     for (j = 0; j &lt; N/3; j++) {         f(N, other);     } }</pre> причем асимптотическая сложность $f(N, other)$ составляет $O(N)$ .<br>Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов) | 1) $O(N^2)$<br>2) $O(N)$<br>3) $O(N \log N)$<br>4) $O(N^2 \log N)$<br>5) $O(N^3)$<br>6) $O(N^3/6)$  | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 10) Принцип организации абстрактного типа данных «стек» (выберите все подходящие варианты ответов)  | 1) FILO (First Input Last Output)<br>2) FIFO (First Input First Output)<br>3) LIFO (Last Input First Output)<br>4) Справедливы варианты 1 и 2 | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 11) Алгоритм сортировка вставками имеет в худшем и лучшем случаях асимптотическую сложность соответственно  | 1) $O(N^2)$ и $O(N^2/2)$<br>2) $O(N^2/2)$ и $O(N^2/4)$<br>3) $O(N^2/2)$ и $O(N)$<br>4) $O(N^2)$ и $O(\ln N)$                                  | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1                       | 12) Какие из следующих алгоритмов имеют асимптотическую сложность $N \log(N)$ в <b>среднем</b> (выберите два подходящие варианта ответов)   | 1) Пирамидальная сортировка<br>2) Сортировка Хоара<br>3) Сортировка вставками<br>4) Сортировка выбором  | средний |

|  |   |   |         |
|--|---|---|---------|
| ОПК-5.2<br>ОПК-5.3   |   |   |         |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 13) В пустое бинарное дерево поиска последовательно добавляются ключи 3, 2, 5, 4. Чему равна разность сумм ключей между левым и правым поддеревьями.            | 1) 5<br>2) 6<br>3) -6<br>4) -7  | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 14) Количество возможных вариантов построения бинарного дерева поиска (его структуры), состоящего из четырех узлов, равно                                       | 1) 12<br>2) 10<br>3) 14<br>4) 18  | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 15) Предложите наиболее оптимальный способ реализации абстрактного типа данных «Множество» (известно, что ключами будут целые числа типа <b>unsigned char</b> ) | 1) упорядоченный список<br>2) упорядоченный массив<br>3) бинарное дерево поиска<br>4) сбалансированное дерево поиска<br>5) хеш-таблица<br>6) битовый массив | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 16) Расположите алгоритмы в порядке повышения производительности (оцениваемой по временной асимптотической сложности в среднем) в <b>среднем</b>                | 1) Бинарный поиск<br>2) Последовательный поиск<br>3) Сортировка вставками<br>4) Сортировка Шелла<br>5) Пирамидальная сортировка                             | высокий |

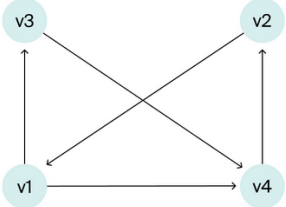
|  |  |   |         |
|--|--|---|---------|
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 17) Расположите алгоритмы в порядке повышения производительности (оцениваемой по асимптотической сложности в среднем по количеству операций сравнения ключей) в <b>среднем</b> | 1) Сортировка Шелла<br>2) Сортировка Хоара<br>3) Сортировка выбором<br>4) Сортировка вставками  | ВЫСОКИЙ |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 18) Расположите алгоритмы в порядке повышения производительности (оцениваемой по асимптотической сложности в среднем по количеству операций сравнения ключей) в <b>среднем</b> | 1) Последовательный поиск<br>2) Интерполяционный поиск<br>3) Поиск прыжками<br>4) Бинарный поиск  | ВЫСОКИЙ |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 19) Пусть есть бинарное дерево, у которого каждый не листовой узел имеет ровно два потомка. Если у такого дерева 11 листьев, то общее количество узлов равно                   | —   | ВЫСОКИЙ |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 20) Предложите наиболее два наиболее оптимальных способа реализации абстрактного типа данных «Множество» (известно, что ключами будут строки)                                  | 1) упорядоченный список<br>2) упорядоченный массив<br>3) бинарное дерево поиска<br>4) сбалансированное дерево поиска<br>5) хеш-таблица<br>6) битовый массив | ВЫСОКИЙ |

**Диагностический тест по дисциплине «Алгоритмы и структуры данных» за четвертый семестр**

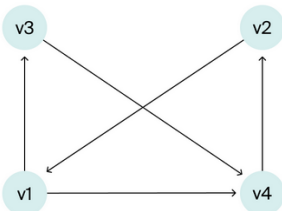
| Проверяемые компетенции  | Задание   | Варианты ответов  | Тип сложности |
|--|---|---|---------------|
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 1) Чтобы алгоритм бинарного поиска работал правильно, нужно, чтобы массив (список) был:                       | 1) Отсортированным<br>2) В куче<br>3) Несортированным<br>4) Выходящим из стека  | низкий        |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 2) Определите максимальное количество узлов в двоичном дереве с высотой $k$ , где корень — нулевая высота (0) | 1) $2^k - 1$<br>2) $2^{k-1} + 1$<br>3) $2^{k+1} - 1$<br>4) $2^k + 1$  | низкий        |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 3) Что означает следующая фраза: «алгоритм $X$ асимптотически более эффективен, чем $Y$ »?                    | 1) $X$ будет лучшим выбором для всех входов<br>2) $X$ будет лучшим выбором для всех входов, за исключением, возможно, небольших входов<br>3) $X$ будет лучшим выбором для всех входов, кроме больших входов<br>4) $Y$ будет лучшим выбором для небольших входов | низкий        |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1                       | 4) Алгоритм обхода графа отличается от алгоритма обхода вершин дерева тем, что...                             | 1) Деревья не соединяются<br>2) Графы могут иметь циклы<br>3) У деревьев есть корни<br>4) Все утверждения выше ошибочны: дерево — подмножество графа  | низкий        |

|  |   |   |         |
|--|---|---|---------|
| ОПК-5.2<br>ОПК-5.3   |   |   |         |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 5) Какой алгоритм из нижеперечисленных будет самым производительным, если дан уже отсортированный массив?   | 1) Сортировка слиянием<br>2) Сортировка вставками<br>3) Быстрая сортировка<br>4) Пирамидальная сортировка   | низкий  |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 6) Какой алгоритм реализует следующая функция на языке C++?<br><pre>int Trial (int a, int b, int c) {     if ((a &gt;= b) &amp;&amp; (c &lt; b))         return b;     else if (a&gt;=b)         return Trial(a, c, b);     else         return Trial(b, a, c); }</pre> | 1) Находит максимальное значение a, b и c<br>2) Находит минимальное значение a, b и c<br>3) Находит среднее число a, b и c<br>4) Ничего из вышеперечисленного | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 7) Какая структура данных обеспечивает эффективность добавление в начало, имеющую сложность O(1) (выберите три подходящие варианта ответов)   | 1) связный список<br>2) стек<br>3) очередь<br>4) дерево   | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2            | 8) Пусть алгоритм представлен следующим псевдокодом:<br><pre>for (i = 0; i &lt; N/2; i++) {     for (j = 0; j &lt; N/3; j++) {         f(N, other);     } }</pre>   | 1) $O(N^2)$<br>2) $O(N)$<br>3) $O(N \log N)$<br>4) $O(N^2 \log N)$<br>5) $O(N^3)$<br>6) $O(N^3/6)$  | средний |

|  |   |  |         |
|--|---|--|---------|
| ОПК-5.3  | <p>причем асимптотическая сложность <math>f(N, other)</math> составляет <math>O(N)</math>. Асимптотическая сложность алгоритма равна (выберите два подходящие варианта ответов)</p>                       |  |         |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 9) Алгоритм Дейкстры основан на:  | 1) Парадигме «разделяй и властвуй»<br>2) Динамическом программировании<br>3) Жадном подходе<br>4) Поиске с возвратом   | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 10) Какие алгоритмы не основан на жадном подходе?   | 1) Алгоритм нахождения кратчайшего пути Дейкстры<br>2) Алгоритм Прима<br>3) Алгоритм Крускала<br>4) Алгоритм Хаффмана<br>5) Алгоритм сортировки слиянием<br>6) Алгоритм быстрой сортировки<br>7) Алгоритм нахождения кратчайшего пути Беллмана-Форда | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3<br>ОПК-5.1<br>ОПК-5.2<br>ОПК-5.3 | 11) Рассмотрите подпрограмму ниже и определите её сложность:<br><pre>void function(int n) {   int i, j, count=0;   for (i=n/2; i &lt;= n; i++)     for (j = 1; j &lt;= n; j = j*2)       count++; }</pre> | 1) $O(\log N)$<br>2) $O(N^2)$<br>3) $O(N^2 \log N)$<br>4) $O(N \log N)$  | средний |
| ПК-2.3<br>ПК-4.3<br>ОПК-2.1<br>ОПК-2.2<br>ОПК-2.3                                  | 12) Какие недостатки присуще методу цепочек для разрешения коллизий хеш-функции?  | 1) Недостатков нет.<br>2) При добавлении элемента нужно выделять дополнительную память, при удалении — освобождать.  | средний |

|  |  |   |                |
|--|--|---|----------------|
| <p>ОПК-5.1<br/>ОПК-5.2<br/>ОПК-5.3</p>   |  | <p>3) Цепочки данных в памяти компьютера расположены не последовательно, а в произвольных местах.</p>   |                |
| <p>ПК-2.3<br/>ПК-4.3<br/>ОПК-2.1<br/>ОПК-2.2<br/>ОПК-2.3<br/>ОПК-5.1<br/>ОПК-5.2<br/>ОПК-5.3</p> | <p>13) Какие недостатки присуще методу открытой адресации для разрешения коллизий хеш-функции?</p>   | <p>1) Недостатков нет, при использовании такого метода можно эффективно проводить операции поиска, вставки и удаления.<br/>2) При увеличении количества ячеек, помеченных ярлыком «deleted», скорость выполнения операций будет замедляться.<br/>3) Данные в памяти компьютера расположены не последовательно, а в произвольных местах.</p> | <p>средний</p> |
| <p>ПК-2.3<br/>ПК-4.3<br/>ОПК-2.1<br/>ОПК-2.2<br/>ОПК-2.3<br/>ОПК-5.1<br/>ОПК-5.2<br/>ОПК-5.3</p> | <p>14) Дан граф:</p>  <p>В каком порядке будут посещены вершины алгоритмом поиска в глубину, если начать из v1?</p> | <p>1) v1, v3, v4, v2<br/>2) v1, v4, v2, v3<br/>3) v1, v2, v3, v4<br/>4) v1, v4, v3, v2</p>  | <p>средний</p> |
| <p>ПК-2.3<br/>ПК-4.3<br/>ОПК-2.1<br/>ОПК-2.2<br/>ОПК-2.3<br/>ОПК-5.1<br/>ОПК-5.2<br/>ОПК-5.3</p> | <p>15) Какой алгоритм обхода в глубину эффективней — тот, что хранит список посещённых вершин, или тот, что хранит массив цветов?</p>  | <p>1) Они одинаково эффективны.<br/>2) Алгоритм, хранящий список посещённых вершин, эффективнее по времени.<br/>3) Алгоритм, хранящий массив цветов, эффективнее по времени.</p>  | <p>средний</p> |
| <p>ПК-2.3<br/>ПК-4.3<br/>ОПК-2.1<br/>ОПК-2.2<br/>ОПК-2.3</p>                                     | <p>16) Дан граф:</p>   | <p>1) v1, v2, v3, v4<br/>2) v1, v3, v4, v2<br/>3) v1, v4, v3, v2<br/>4) v1, v3, v4, v2, v1</p>  | <p>высокий</p> |

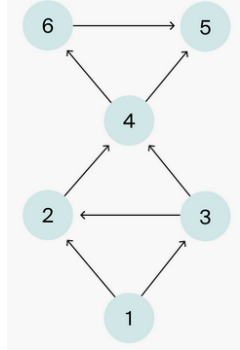
ОПК-5.1  
ОПК-5.2  
ОПК-5.3



Каким будет порядок обхода для алгоритма поиска в ширину, если начать с вершины v1?

ПК-2.3  
ПК-4.3  
ОПК-2.1  
ОПК-2.2  
ОПК-2.3  
ОПК-5.1  
ОПК-5.2  
ОПК-5.3

17) Каким может быть порядок обхода BFS для этого графа при стартовой вершине 1?

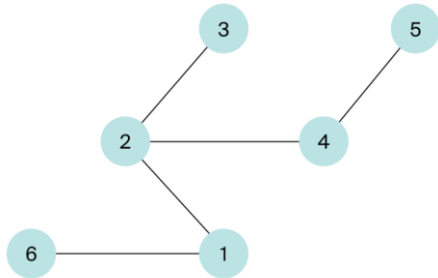


- 1) [1, 2, 3, 4, 6, 5]
- 2) [1, 2, 3, 4, 5, 6]
- 3) [1, 3, 2, 6, 5, 4]
- 4) [1, 3, 2, 4, 5, 6]
- 5) [1, 3, 2, 4, 6, 5]
- 6) [1, 3, 2, 5, 4, 6]

высокий

ПК-2.3  
ПК-4.3  
ОПК-2.1  
ОПК-2.2  
ОПК-2.3  
ОПК-5.1  
ОПК-5.2  
ОПК-5.3

18) Каким может быть порядок обхода BFS для этого графа при стартовой вершине 1?



- 1) [1, 2, 3, 4, 5, 6]
- 2) [1, 2, 3, 4, 6, 5]
- 3) [1, 2, 6, 3, 4, 5]
- 4) [1, 2, 6, 4, 3, 5]
- 5) [1, 6, 2, 4, 3, 5]
- 6) [1, 6, 2, 4, 5, 3]

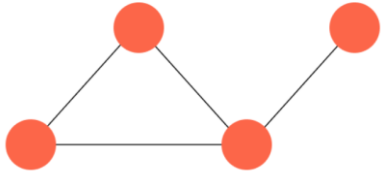
высокий

ПК-2.3  
ПК-4.3

19) Сколько разных остовных деревьев у этого графа?

—

высокий

|   |  |   |                |
|---|--|---|----------------|
| <p>ОПК-2.1<br/> ОПК-2.2<br/> ОПК-2.3<br/> ОПК-5.1<br/> ОПК-5.2<br/> ОПК-5.3</p>                         |   |   |                |
| <p>ПК-2.3<br/> ПК-4.3<br/> ОПК-2.1<br/> ОПК-2.2<br/> ОПК-2.3<br/> ОПК-5.1<br/> ОПК-5.2<br/> ОПК-5.3</p> | <p>20) Если поиск и извлечение минимального по весу ребра происходит из массива, то какая у алгоритма Прима сложность?</p> | <p>1) <math>O( V ^2 \log E )</math><br/> 2) <math>O( V  \cdot  E )</math><br/> 3) <math>O( V  +  E )</math><br/> 4) <math>O( E  +  V  \log V )</math></p> | <p>высокий</p> |