Бабенко Максим Александрович,
Станкевич Андрей Сергеевич

ТЕСТ ПО «COMPUTER SCIENCE»

ВВЕДЕНИЕ

Каждый год зимой и летом в России проводятся учебно-тренировочные сборы по информатике. Целью этих сборов является формирование и подготовка сборной команды школьников России по информатике на международную олимпиаду IOI [16]. Преподавателями на сборах обычно являются студенты и аспиранты различных вузов страны, в основном МГУ и СПбГИТМО (ТУ), в частности, в состав жюри последних сборов входили и авторы этой статьи.

Следует отметить, что уровень сложности теоретических лекций и задач, которые предлагаются на сборах, традиционно достаточно высок. Задачи и материалы лекций последних сборов можно найти на сайтах [17] и [18]. Также большое количество задач с различных сборов можно найти в замечательной книге [1].

Поскольку в последние годы состав участников сборов существенно обновился, появилась необходимость в выяснении уровня знаний школьников, приглашенных на сборы, с целью составления и дальнейшей корректировки плана лекций. Для этого авторами данной статьи был составлен тест, который и приведен ниже. Результаты теста послужили ориентиром, показывающим необходимость освещения на лекциях тех или иных тем.

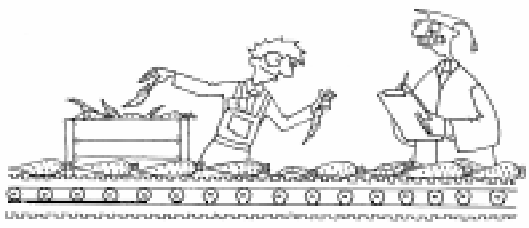
Предлагаем и Вам попробовать свои силы и ответить на вопросы теста. Стоит заметить, что в тесте есть как очень простые, так и весьма сложные вопросы. Большинство участников сборов успешно справилось лишь с 20–30% вопросов. И лишь наиболее сильные школьники, которые в результате и составили сборную команду страны, ответили правильно на большинство вопросов (отметим, однако, что результаты теста на отбор в команду напрямую не влияли, просто корреляция между результатом теста и результатом, показанным школьником на сборах, была достаточно хорошо заметна).

Особо следует выделить результат трехкратного чемпиона России, двукратного медалиста международных олимпиад и члена команды-победителя последней Всероссийской командной олимпиады школьников Петра Митричева. Он не просто дал правильные ответы на подавляющее большинство вопросов теста, но и помог членам жюри исправить некоторые неточности, допущенные ими в первоначальном варианте теста.

Так что свои результаты следует оценивать лояльно. Желающих же подробнее ознакомиться с темами, поднятыми в этом тесте, отсылаем к книгам и ссылкам, приведенным в списке литературы.

ТЕСТ

СОРТИРОВКА И ПОИСК



1. Заполните таблицу параметров различных сортировок (N – число элементов)

Алгоритм	T_{max}	M_{max}	C_{max}
Сортировка пузырьком (bubble sort)			
Сортировка слиянием (merge sort)			
Сортировка пирамидой (heap sort)			
Быстрая сортировка (quick sort)			

Примечание: T_{max} – максимальное время работы, M_{max} – максимальный объем дополнительной памяти, C_{max} – максимальное число сравнений. Везде достаточно указать только главный член асимптотики с точностью до константы, например C_{max} есть N^2 для сортировки пузырьком.

2. Первоначальное построение пирамиды в алгоритме heapsort производится за время

☐ $\log n$ ☐ $n^{1/2}$ ☐ n
☐ $n \log n$ ☐ n^2

3. Процедура «просеивания» (sift) в алгоритме heapsort производится за время

☐ $\log n$ ☐ $n^{1/2}$ ☐ n
☐ $n \log n$ ☐ n^2

4. Пусть массив a [1..n] содержит пирамиду. Тогда предок вершины с номером i – это вершина с номером

☐ 1 ☐ i ☐ $i - 1$
☐ $i + 1$ ☐ i *div* 2 ☐ (i *div* 2) + 1

5. По определению массив a [1..n] содержит пирамиду, если

☐ $a[i]$ – все различные
☐ $a[1] \le a[2] \le ... \le a[n]$
☐ $a[i] \ge a[2*i]$, $a[i] \ge a[2*i+1]$ при всех допустимых i
☐ $a[2*i] \ge a[2*i + 1]$ при всех допустимых i

ТЕОРИЯ ГРАФОВ

6. Критерием наличия в графе цикла отрицательного веса при применении к нему алгоритма Флойда служит

☐ зацикливание алгоритма
☐ завершение алгоритмом работы
☐ обнуление матрицы оценок расстояний
☐ появление в матрице отрицательных значений
☐ появление на диагонали матрицы отрицательных значений

7. Заполните таблицу параметров различных алгоритмов на графах (N – число вершин в графе, M – число ребер).

Алгоритм	*SS*	W^+	T
алгоритм Флойда			
алгоритм Форда-Беллмана			
алгоритм Дейкстры (простейший вариант)			
алгоритм Дейкстры (с пирамидой)			
алгоритм Дейкстры (с фибоначчиевыми кучами)			

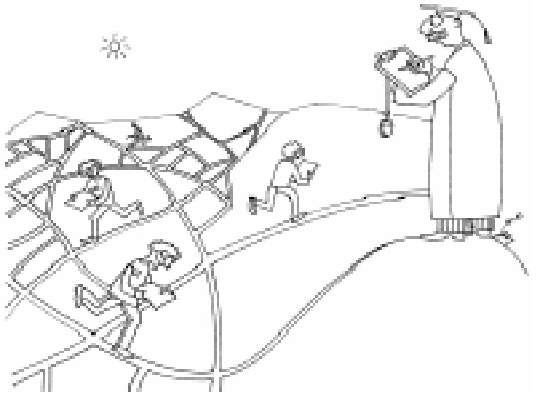
Примечание: SS (single source) – поиск расстояний от фиксированной вершины (да/нет), W^+– требование неотрицательности весов ребер (да/нет), T – время работы (главный член асимптотики с точностью до константы).

8. После k итераций внешнего цикла алгоритма Флойда элемент $a[i, j]$ матрицы расстояний содержит

☐ кратчайшую длину пути от вершины i до вершины j
☐ кратчайшую длину пути, состоящего не более чем из k ребер, от вершины i до вершины j
☐ кратчайшую длину пути, проходящего через вершины с номерами не более k, от вершины i до вершины j

9. После k итераций внешнего цикла алгоритма Форда-Беллмана элемент $d[i]$ массива расстояний содержит

☐ кратчайшую длину пути до вершины i



☐ кратчайшую длину пути, состоящего не более чем из k ребер, до вершины i
☐ кратчайшую длину пути, проходящего через вершины с номерами не более k, до вершины i

10. Можно ли применить (слегка модифицированный) алгоритм Дейкстры для поиска наименьшего по весу пути из одной вершины во все остальные, если весом пути считается
☐ произведение весов ребер, веса ребер неотрицательны
☐ произведение весов ребер, веса ребер ≥ 1
☐ вес наименьшего ребра в пути
☐ вес наибольшего ребра в пути
☐ сумма величин, обратных весам ребер, веса ребер положительны
☐ корень квадратный из суммы квадратов весов ребер
☐ среднее арифметическое весов ребер, веса ребер неотрицательны
☐ среднее геометрическое весов ребер, веса ребер положительны

11. Рассмотрим следующий алгоритм, считая, что в матрице $a[1..n, 1..n]$ заданы веса ребер взвешенного ориентированного графа с n вершинами, где некоторые символы заменены знаками вопроса. На какие символы следует заменить эти знаки вопроса, чтобы получился алгоритм Флойда?

```
for i = 1..n do
 for j = 1..n do
 for k = 1..n do
  a[?][?] <- min(a[?][?], a[?][?] + a[?][?])
```

☐ `ijijikkj` ☐ `ijijikjk`
☐ `jkjkjiik` ☐ `jkjkjiki`
☐ `ikikijjk` ☐ `ikikijkj`

12. Выберите верные утверждения (ОГ – ориентированный граф, НОГ– неориентированный граф)
☐ В НОГ с N вершинами не менее $N-1$ ребер
☐ В дереве с N вершинами не менее $N-1$ ребер
☐ Если в НОГ N вершин и $N-1$ ребер, то он является деревом
☐ Если в ОГ существует отрицательный путь, то в нем существует отрицательный цикл
☐ Если в НОГ существует отрицательный путь, то в нем существует отрицательный цикл
☐ Если в ОГ существует отрицательный путь, то в нем существует отрицательный простой путь
☐ Если в НОГ существует отрицательный путь, то в нем существует отрицательный простой путь
☐ Если в ОГ существует отрицательный цикл, то в нем существует отрицательный простой цикл
☐ Если в НОГ существует отрицательный цикл, то в нем существует отрицательный простой цикл
☐ Если в ОГ с N вершинами существует отрицательный цикл, то в нем существует отрицательный цикл, содержащий не более N ребер
☐ Если в НОГ с N вершинами существует отрицательный цикл, то в нем существует отрицательный цикл, содержащий не более N ребер

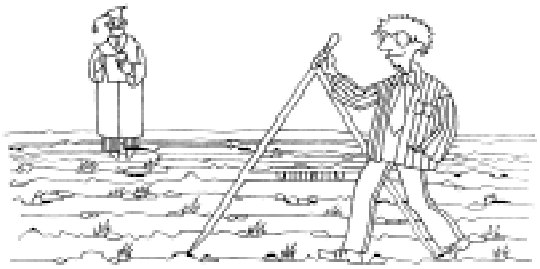
13. Рассмотрим следующие утверждения про неориентированный граф с числом вершин $N \geq 3$:
(a) граф двудольный
(b) граф не имеет циклов
(c) все циклы графа имеют четную длину
(d) граф является деревом
(e) граф является лесом
(f) граф связен
(g) граф является полным
(h) граф содержит N вершин и $N-1$ ребро
(i) граф не содержит ребер
(j) в графе из любой вершины в любую существует путь
(k) в графе одновременно выполняются (f) и (h)
(l) в графе существует ровно один простой путь из любой вершины в любую

Заполните таблицу 1, поставив крестик в позиции (i, j) (i – номер строки, j – столбца), если из i-го утверждения следует j-е. В окне справа нарисуйте ориентированный граф, для которого таблица слева является матрицей смежности. Постарайтесь, чтобы жюри могло сразу понять, что Ваш граф является верным.

Таблица 1.

	a	b	c	d	e	f	g	h	i	j	k	l
a												
b												
c												
d												
e												
f												
g												
h												
i												
j												
k												
l												

ВЫЧИСЛИТЕЛЬНАЯ ГЕОМЕТРИЯ



14. Ориентированная площадь параллелограмма, натянутого на векторы (x_1, y_1) и (x_2, y_2) равна

☐ 0 ☐ $x_1y_1 + x_2y_2$
☐ $x_1y_1 - x_2y_2$ ☐ $x_1y_2 - x_2y_1$
☐ $x_1y_2 + x_2y_1$

15. Пусть на плоскости расположено N точек, причем известно, что их выпуклая оболочка – не более чем 10-угольник. Тогда для ее построения более эффективно использовать алгоритм

☐ Дейкстры ☐ Джарвиса
☐ Флойда ☐ Форда-Беллмана
☐ Грэхема

16. Наиболее медленной частью алгоритма Грэхема является

☐ обход выпуклой оболочки с использованием стека
☐ вычисление полярных углов
☐ поиск кратчайших расстояний
☐ сортировка полярных углов
☐ построение транзитивного замыкания
☐ взлом криптосистемы RSA

17. Пусть прямая задана уравнением $Ax + By + C = 0$, тогда расстояние от точки (x_0, y_0) до данной прямой равно

☐ $\left|Ax_0 + By_0\right|$
☐ $\left|Ax_0 + By_0 + C\right|$
☐ $\left|Ax_0 + By_0 + C\right| \Big/ \sqrt{A^2 + B^2}$
☐ $\left|Ax_0 + By_0 + C\right| \Big/ \sqrt{A^2 + B^2 + C^2}$
☐ $\left|Ax_0 + By_0\right| \Big/ \sqrt{A^2 + B^2 + C^2}$
☐ $\left|Ax_0 + By_0\right| \Big/ \sqrt{A^2 + B^2}$

ТЕОРИЯ КОНЕЧНЫХ АВТОМАТОВ И КОНТЕКСТНО-СВОБОДНЫХ ГРАММАТИК

CFG = context-free grammar, DFA = deterministic finite automaton, NFA = non-deterministic finite automaton

18. Язык $\{a^n : n \geq 0\}$ является
- ☐ регулярным, но не контекстно-свободным
- ☐ контекстно-свободным, но не регулярным
- ☐ контекстно-свободным и регулярным
- ☐ не контекстно-свободным и не регулярным

19. Язык $\{a^n b^n : n \geq 0\}$ является
- ☐ регулярным, но не контекстно-свободным
- ☐ контекстно-свободным, но не регулярным
- ☐ контекстно-свободным и регулярным
- ☐ не контекстно-свободным и не регулярным

20. Язык $\{a^n b^n c^n : n \geq 0\}$ является
- ☐ регулярным, но не контекстно-свободным
- ☐ контекстно-свободным, но не регулярным
- ☐ контекстно-свободным и регулярным
- ☐ не контекстно-свободным и не регулярным

21. Доказательство теоремы о регулярности каждого автоматного языка использует прием, аналогичный шагу алгоритма
- ☐ Дейкстры
- ☐ Джарвиса
- ☐ Флойда
- ☐ Форда-Беллмана
- ☐ Грэхема

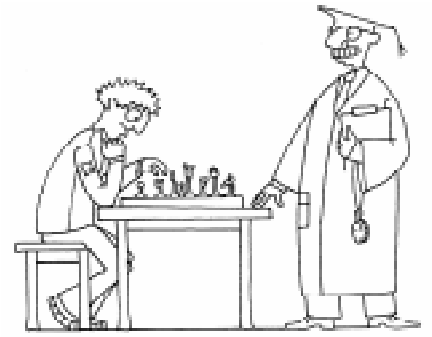
22. Пусть NFA имеет n состояний, тогда соответствующий ему DFA всегда имеет не более
- ☐ 1
- ☐ n
- ☐ 2^n
- ☐ n^2
- ☐ $n!$
- состояний

23. Алгоритм Кнута-Морриса-Пратта ищет подстроку длины M в строке длины N за время
- ☐ MN
- ☐ $M \log N$
- ☐ $N \log M$
- ☐ $M + N$

24. Выберете верные свойства алгоритма построения ε-замыкания (ε-closure) NFA
- ☐ результатом работы в общем случае является NFA
- ☐ результатом работы всегда является DFA
- ☐ в полученном автомате нет ε-переходов
- ☐ в полученном автомате из вершины не может идти пара ребер с одинаковыми пометками
- ☐ в полученном автомате из вершины может идти пара ребер с одинаковыми пометками

25. Идея, лежащая в основе алгоритма Рабина-Карпа поиска подстрок
- ☐ ρ-эвристика Полларда
- ☐ хеширование
- ☐ кэширование
- ☐ линеаризация
- ☐ интерполяция
- ☐ экстраполяция
- ☐ объектно-ориентированное проектирование

КОМБИНАТОРИКА



26. Последовательность Фибоначчи начинается так:
- ☐ 1, 2, 3, 4...
- ☐ 1, 4, 9, 16...
- ☐ 1, 3, 5, 7...
- ☐ 1, 1, 2, 3...
- ☐ 2, 3, 5, 7...

27. В правильной скобочной последовательности n открывающих скобок. Тогда число закрывающих скобок равно:
- ☐ невозможно определить
- ☐ $n/2$
- ☐ $2n$
- ☐ n
- ☐ $n + 1$
- ☐ $n - 1$

28. Число правильных скобочных последовательностей из n пар скобок равно
- ☐ 2^n
- ☐ n-ому числу Фибоначчи
- ☐ $(n - 1)$-ому числу Фибоначчи
- ☐ $(n + 1)$-ому числу Фибоначчи
- ☐ n-ому числу Каталана

☐ $(n-1)$-ому числу Каталана
☐ $(n+1)$-ому числу Каталана
☐ $n!$
☐ n-ому числу Кармайкла

29. Число последовательностей из n нулей и единиц без двух единиц подряд равно
☐ 2^n
☐ n-ому числу Фибоначчи
☐ $(n+1)$-ому числу Фибоначчи
☐ $(n+2)$-ому числу Фибоначчи
☐ n-ому числу Каталана
☐ $(n+1)$-ому числу Каталана
☐ $(n+2)$-ому числу Каталана
☐ $n!$
☐ n-ому числу Кармайкла

30. n-е число Каталана равно
☐ $n!$ ☐ $\dfrac{(2n)!}{(n!)^2}$ ☐ $\dfrac{(2n)!}{(n+1)(n!)^2}$ ☐ $\dfrac{(2n)!}{n!}$

31. Формула Кэли утверждает, что число различных помеченных деревьев на N вершинах равно _____

32. Тождественная перестановка на n элементах является
☐ четной ☐ нечетной
☐ зависит от n

33. Пусть имеется перестановка из n элементов и $O(1)$ дополнительной памяти. Какие из приведенных алгоритмов позволяют вычислить ее четность за время $O(n \log n)$?
☐ сортировка пузырьком (bubble sort)
☐ сортировка пирамидой (heap sort)
☐ сортировка слиянием (merge sort)
☐ быстрая сортировка (quick sort)

34. Декремент перестановки на n элементах равен
☐ числу независимых циклов (включая циклы длины 1)
☐ числу неподвижных точек
☐ n – число независимых циклов (включая циклы длины 1)
☐ n – число неподвижных точек

35. Наименьшее число транспозиций, на которое можно разложить перестановку, равно
☐ числу независимых циклов (включая циклы длины 1)
☐ декременту
☐ числу неподвижных точек
☐ перманенту

36. Рассмотрим формулу включения-исключения для n множеств. Число (нетривиальных) слагаемых в альтернированной сумме равно
☐ $n!$ ☐ n ☐ 2^n ☐ $2^n - 1$

ТЕОРЕТИКО-ЧИСЛОВЫЕ И ПОЛУЧИСЛЕННЫЕ АЛГОРИТМЫ

37. Пусть сравнение $a^{n-1} \equiv 1 \pmod n$ (тест малой теоремы Ферма) для модуля n дал положительный ответ для всех оснований от 1 до $n-1$. Тогда можно утверждать, что
☐ ничего нельзя утверждать
☐ такого не бывает
☐ n – простое или число Кармайкла
☐ n – составное
☐ n – число Кармайкла
☐ n – число Ферма
☐ n – простое число Ферма
☐ n – простое

38. Эвристическая оценка времени работы ρ-эвристики Полларда есть
☐ $\log n$ ☐ n ☐ $n^{1/2}$
☐ $n^{1/4}$ ☐ $n^{1/8}$

39. Числа Кармайкла являются
☐ простыми
☐ составными

☐ существует единственное такое число, и оно равно единице

40. Для проверки числа n на простоту достаточно проверить его остатки от деления на числа до
☐ $\log n$ ☐ 10! ☐ $n^{1/3}$ ☐ $n^{1/2}$

41. Ненулевой вычет a обратим по модулю m, если и только если
☐ всегда обратим ☐ m – простое
☐ m не делится на a ☐ a – простое
☐ m и a взаимно просты

42. Асимптотический закон распределения простых чисел утверждает, что число простых чисел, не превосходящих n, асимптотически равно
☐ $n/2$ ☐ $n^{1/2}$ ☐ $n^{1/3}$
☐ $n / \ln n$ ☐ $n / \ln \ln n$

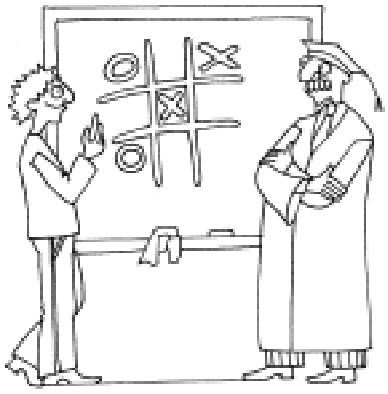
43. Наиболее быстрый из известных алгоритмов позволяет умножать двоичные n-разрядные числа за
☐ n ☐ n^2
☐ $n!$ ☐ $n \log n$
☐ $n \log n \log \log n \log \log \log n$
☐ $n \log n \log \log n$ битовых операций.

44. Выберите верные утверждения (ПФ = преобразование Фурье; ОПФ = обратное ПФ):
☐ свертка последовательностей равна их поэлементному произведению
☐ свертка последовательностей равна ПФ от их поэлементного произведения
☐ ПФ от свертки последовательностей равно поэлементному произведению их ОПФ
☐ свертка последовательностей равна ОПФ от поэлементного произведения их ПФ

45. Быстрое преобразование Фурье (БПФ) последовательности требует времени
☐ n ☐ n^2 ☐ n^3
☐ n^4 ☐ $n^2 \log n$ ☐ $n \log n$

46. Критерием разрешимости в целых числах уравнения $ax + by = c$ является условие
☐ данное уравнение всегда разрешимо
☐ *НОД* (a, b) делит c
☐ c делит *НОД* (a, b)
☐ $c = 1$
☐ *НОД* $(a, b) = 1$
☐ *НОД* $(a, b, c) = 1$
☐ *НОД* $(a, b) = c$

ЛИНЕЙНАЯ АЛГЕБРА И ТЕОРИЯ МАТРОИДОВ



47. Время работы алгоритма Гаусса решения систем n линейных уравнений есть
☐ $\log n$ ☐ n ☐ n^2
☐ n^3 ☐ n^4

48. Выбор максимального по модулю элемента в качестве ведущего в алгоритме Гаусса применяется для
☐ ускорения работы
☐ упрощения алгоритма
☐ улучшения точности

49. Пусть в связном графе n вершин и m ребер. Тогда максимальное значение, которое может принимать ранговая функция его циклического матроида, равно
☐ 0 ☐ 1 ☐ m
☐ n ☐ $m – n$ ☐ $n – m$
☐ $m – 1$ ☐ $n – 1$ ☐ $min\ (n, m)$
☐ $max\ (n, m)$

50. Выберите верные утверждения:
☐ ранг объединения множеств есть сумма рангов слагаемых
☐ ранг объединения множеств не больше суммы рангов слагаемых
☐ подмножество базы есть база
☐ подмножество независимого множества есть независимое множество
☐ все базы матроида равномощны
☐ все независимые множества матроида равномощны

51. Поиск наибольшего паросочетания в двудольном графе сводится к задаче

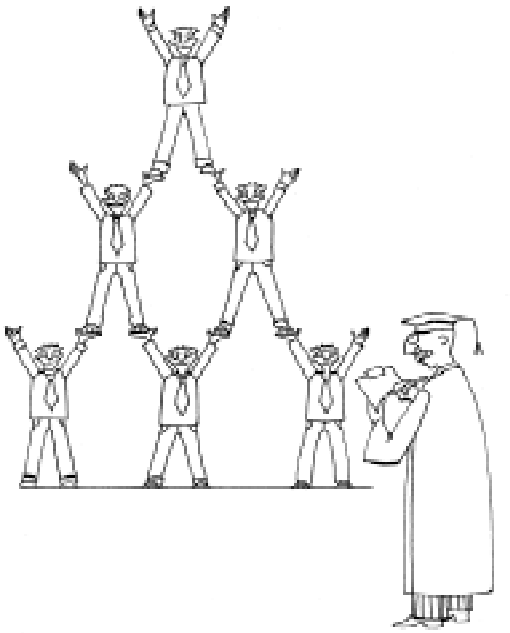
☐ поиска максимального независимого множества в матроиде
☐ пересечения двух матроидов
☐ объединения двух матроидов

52. Какие из следующих семейств множеств образуют семейство независимых множеств матроида?
☐ всевозможные подмножества конечного множества
☐ всевозможные непустые подмножества конечного множества
☐ ациклические подмножества ребер ориентированного графа
☐ ациклические подмножества ребер неориентированного графа
☐ всевозможные паросочетания двудольного графа
☐ всевозможные частичные трансверсали семейства множеств
☐ линейно-независимые подмножества конечного множества элементов линейного пространства
☐ линейно-зависимые подмножества конечного множества элементов линейного пространства

53. Пусть в процессе применения градиентного алгоритма (теорема Радо-Эдмондса) к матроиду было построено независимое множество, и к нему невозможно добавить больше элементов без нарушения свойства независимости. Это означает, что
☐ этого не может быть
☐ алгоритм зациклится
☐ алгоритм закончит работу
☐ построена база экстремального веса
☐ построена антибаза экстремального веса
☐ алгоритм не применим к данному матроиду

54. Рассмотрим матроид строк матрицы размера $M \times N$. Тогда его ранг равен
☐ M
☐ N
☐ max (N, M)
☐ min (N, M)
☐ рангу матрицы
☐ определителю матрицы
☐ перманенту матрицы

СТРУКТУРЫ ДАННЫХ



55. Структура пирамиды (heap) используется для организации
☐ стека (stack)
☐ очереди (queue)
☐ дека (deque)
☐ приоритетной очереди (priority queue)
☐ сбалансированного дерева (balanced tree)

56. Вращения сбалансированного дерева применяются для
☐ изменения порядка вершин дерева при естественном обходе
☐ восстановления сбалансированности
☐ не применяются
☐ для увеличения глубины дерева

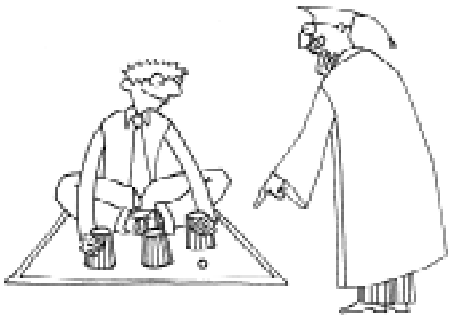
57. Возможность применять вращения к дереву основана на свойстве, схожем с алгебраическим свойством
☐ коммутативности
☐ ассоциативности
☐ дистрибутивности
☐ идемпотентности

58. Выберите верные свойства красно-черного дерева с N вершинами
☐ глубина дерева всегда не превосходит $\log N + 1$
☐ если вершина красная, то оба ее потомка красные
☐ если вершина красная, то оба ее потомка черные

☐ все пути от корня имеют одинаковую длину
☐ все пути от корня имеют одинаковую красную длину
☐ все пути от корня имеют одинаковую черную длину
☐ каждый лист (*NIL*) дерева черный
☐ каждый лист (*NIL*) дерева красный

59. Выберите задачи, для решения которых пригодна структура данных «система непересекающихся множеств».
☐ поиск кратчайших путей в графе
☐ поиск связных компонент графа
☐ поиск двусвязных компонент графа
☐ поиск минимального остова графа
☐ поиск минимального вершинного покрытия графа

ЭЛЕМЕНТЫ ТЕОРИИ ВЕРОЯТНОСТЕЙ



60. Найдите вероятность следующих событий при бросании честной монеты, на которой равновероятно выпадает 0 или 1 (далее просто *«честная монета»*).
- при бросании честной монеты выпадает 1: _____
- при двух бросаниях выпадает две единицы подряд: _____
- при втором бросании выпадает 1 при условии, что при первом бросании выпало 1: _____
- при N бросаниях выпадет ровно K единиц: _____
- при $2N-1$ бросании выпадет как минимум N единиц: _____
- при $2N$ бросаниях выпадет как минимум N единиц: _____

61. Найдите вероятность следующих событий при бросании нечестной монеты, на которой выпадает 0 или 1, причем 0 с вероятностью p (далее просто *«нечестная монета»*).
- выпадает 1: _____
- при двух бросаниях выпадают две единицы подряд: _____
- при втором бросании выпадает 1 при условии, что при при первом бросании выпала 1: _____
- при N бросаниях выпадет ровно K единиц: _____

62. Найдите математическое ожидание следующих случайных величин
- значение при одном броске честной монеты: _____
- значение при одном броске нечестной монеты: _____
- сумма при броске N честных монет: _____
- количество бросаний честной монеты, за которое будет получена 1: _____
- выигрыш в следующей игре: я плачу \$100 и бросаю честную монету, если 1 выпадает впервые на N-ом шаге, я получаю \$$N$: _____
- выигрыш в следующей игре: я плачу \$100 и бросаю честную монету, если 1 выпадает впервые на N-ом шаге, я получаю \$$N^2$: _____
- выигрыш в следующей игре: я плачу \$100 и бросаю честную монету, если 1 выпадает впервые на N-ом шаге, я получаю \$$2^N$: _____

63. Выберите верные утверждения, если x и y – независимые случайные величины, c – константа (E – символ математического ожидания, D – дисперсии).
☐ $E(x+y) = E(x) + E(y)$
☐ $E(xy) = E(x)E(y)$
☐ $E(cx) = cE(x)$
☐ $E(cx) = c^2E(x)$
☐ $E(cx) = E(x)/c$
☐ $E(cx) = E(x)$
☐ $D(x) = E(x - E(x))$
☐ $D(x) = E(x)^2 - E(x^2)$
☐ $D(x) = E(x^2) - E(x)^2$
☐ $D(x+y) = D(x) + D(y)$
☐ $D(xy) = D(x)\ D(y)$
☐ $D(cx) = cD(x)$

☐ $D(cx) = c^2D(x)$
☐ $D(cx) = D(x)/c$
☐ $D(cx) = D(x)$
☐ $P(x = a, y = b) = P(x = a) + P(y = b)$
☐ $P(x = a, y = b) = P(x = a)\ P(y = b)$
☐ $P(x + y = a + b) = P(x = a) + P(y = b)$
☐ $P(x + y = a + b) = P(x = a)\ P(y = b)$

64. Выберите верные утверждения (считайте, что все математические ожидания и дисперсии существуют).
☐ $E(xy) \le E(x)\ E(y)$
☐ $E(xy)^2 \le E(x^2)\ E(y^2)$
☐ $E(xy)^2 \le E(x)^2\ E(y)^2$
☐ $P(|x| \ge L) \le E(|x|) / L$
☐ $P(|x| \ge L) \le E(x^2) / L^2$
☐ $P(|x| \ge L) \le E(|x|^k) / L^k$, k – натуральное
☐ $P(|x - E(x)| \ge L) \le D(x)/L^2$, $L>0$
☐ $P(|x - E(x)| \ge L) \le D(x)^2/L$, $L>0$
☐ $P(|x - E(x)| \ge L) \le D(x)^2/L^2$, $L>0$
☐ $P(|x| \ge L) \le E(|x|^{1/2})/L^{1/2}$, $L>0$
☐ $P(|x - E(x)| \ge L) \le D(x)/L$, $L>0$

ЭЛЕМЕНТЫ ТЕОРИИ СЛОЖНОСТИ И КРИПТОГРАФИИ

Сложностные классы: P = polynomial time, NP = non-deterministic polynomial time, PSPACE = polynomial space, ZKP = zero knowledge proof, IP = interactive proof, AM = Arthur-Merlin games, BPP = bounded probability polynomial, RP = one-sided error randomized polynomial, ZPP = zero-error probability polynomial.

65. Отметьте все NP-полные языки в списке
☐ множество эйлеровых графов
☐ множество гамильтоновых графов
☐ множество пар (G, k), где G – граф, в котором существует простой путь длины k
☐ множество пар изоморфных графов
☐ множество пар (G, k), где G – граф, в котором есть вершинное покрытие мощности k
☐ множество пар (G, k), где G – двудольный граф, в котором есть вершинное покрытие мощности k
☐ множество пар (G, k), где G – граф, в котором есть паросочетание мощности k
☐ множество пар (G, k), где G – двудольный граф, в котором есть паросочетание мощности k
☐ множество выполнимых 2-КНФ (2-SAT problem)
☐ множество выполнимых 3-КНФ (3-SAT problem)
☐ множество пар (Y, a), где Y – задача линейного программирования, чей оптимум не превосходят a (linear programming problem)
☐ множество пар (Y, a), где Y – задача дискретного программирования, чей оптимум не превосходят a (discrete programming problem)
☐ множество истинных булевых формул с кванторами (truth quantified Boolean formula)
☐ множество машин Тьюринга, останавливающихся на пустом входе (halting problem)

66. Наличие теста Миллера-Рабина на простоту непосредственно показывает, что язык PRIMES, задающий множество простых чисел, принадлежит следующим сложностным классам (выберите все правильные ответы):

☐ P	☐ NP	☐ co-NP
☐ ZKP	☐ IP	☐ AM
☐ BPP	☐ RP	☐ co-RP
☐ ZPP		

67. Язык называется NP-трудным (NP-hard), если
☐ к нему сводятся по Карпу все языки из P
☐ к нему сводятся по Карпу все языки из NP
☐ к нему сводятся по Карпу все языки из NP, но сам он в P не лежит
☐ к нему сводятся по Карпу все языки из NP, но сам он в NP не лежит

☐ к нему сводятся по Карпу все языки из NP, и сам он лежит в P
☐ к нему сводятся по Карпу все языки из NP, и сам он лежит в NP

68. Язык называется NP-полным (NP-complete), если
☐ к нему сводятся по Карпу все языки из P
☐ к нему сводятся по Карпу все языки из NP
☐ к нему сводятся по Карпу все языки из NP, но сам он в P не лежит
☐ к нему сводятся по Карпу все языки из NP, но сам он в NP не лежит
☐ к нему сводятся по Карпу все языки из NP, и сам он лежит в P
☐ к нему сводятся по Карпу все языки из NP, и сам он лежит в NP

69. Надежность криптосистемы RSA непосредственно основана на (недоказанной) сложности задачи
☐ поиска кратчайшего пути (shortest-path problem)
☐ дискретного логарифмирования по простому модулю (discrete logarithm problem)
☐ целочисленной факторизации (integer factorization problem)
☐ упаковки рюкзака (knapsack problem)

70. Пусть предложенную жюри сборов задачу X удалось свести к NP-полной задаче Y. Это означает
☐ такого не бывает
☐ исходная задача X, скорее всего, не имеет полиномиального алгоритма решения
☐ выбранный метод сведения неудачен, следует попробовать найти другой
☐ исходная задача заведомо имеет полиномиальный алгоритм решения

71. Пусть к предложенной жюри сборов задаче X удалось свести NP-полную задачу Y. Это означает
☐ такого не бывает
☐ исходная задача X, скорее всего, не имеет полиномиального алгоритма решения
☐ выбранный метод сведения неудачен, следует попробовать найти другой
☐ исходная задача заведомо имеет полиномиальный алгоритм решения

72. Предположим, что P = NP. Выберите все верные следствия.
☐ не существует односторонних функций
☐ задача дискретного логарифмирования разрешима за полиномиальное время
☐ задача остановки машины Тьюринга разрешима за полиномиальное время
☐ задача определения истинности булевых формул с кванторами (truth quantified Boolean formula) разрешима за полиномиальное время
☐ существуют односторонние функции

73. Рассмотрим криптосистему RSA ($n = pq$, p, q – простые, x – сообщение, e – открытый ключ, d – секретный ключ). Выберите верные утверждения.
☐ знание (n, p, q, e) позволяет найти d
☐ знание (n, e, d) позволяет найти p и q
☐ знание $(n, e, x^e \bmod n)$ позволяет найти x
☐ знание $(n, d, x^e \bmod n)$ позволяет найти x
☐ знание $(n, d, x^e \bmod n)$ позволяет найти e
☐ знание $(n, d, x^e \bmod n)$ позволяет найти p и q

Примечание: всюду здесь и далее имеется в виду существование детерминированного или рандомизированного алгоритма, работающего полиномиальное время.

74. Пусть $a^x \bmod p = b$, p – простое. Тогда для вычисления x по известным (a, b, p) достаточно
☐ $O(p)$ времени и $O(1)$ памяти
☐ $O(p^{1/2})$ времени и $O(p^{1/2})$ памяти
☐ $O(\log p)$ времени и $O(p)$ памяти
☐ $O(1)$ времени и $OT(p)$ памяти

Примечание: считаем, что арифметические операции над числами по модулю p требуют времени $O(1)$, равно как и хранение вычета по данному модулю – памяти $O(1)$.

75. Пусть, как и системе RSA, $n = pq$, где p, q – простые, $a^2 \bmod n = b$. Тогда
☐ наличие эффективного алгоритма для взлома криптосистемы RSA позволяет по известным (b, n) эффективно находить a
☐ наличие эффективного алгоритма для вычисления дискретного логарифма по простому модулю позволяет по известным (b, n) эффективно находить a
☐ знание n и возможность полиномиальное число раз спрашивать у оракула

значения a для любых b позволяет за полиномиальное время и с вероятностью не ниже 1/2 найти p и q

- ☐ знание (b, n) позволяет найти a
- ☐ знание (b, p, q) позволяет найти a
- ☐ знание (a, b, n) позволяет найти p и q

Литература

1. *Беров В., Лапунов А., Матюхин В., Пономарев А.* Особенности национальных задач по информатике.
2. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ.
3. *Кнут Д.* Искусство программирования, том 2. Получисленные алгоритмы.
4. *Кнут Д.* Искусство программирования, том 3. Сортировка и поиск.
5. *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики.
6. *Ахо А., Сети Р., Ульман Д.* Компиляторы: принципы, технологии и инструменты.
7. *Ахо А., Ульман Д.* Теория синтаксического анализа, перевода и компиляции.
8. *Ахо А., Ульман Д., Хопкрофт Д.* Построение и анализ вычислительных алгоритмов.
9. Введение в криптографию / Под общ. ред. *В. Ященко*.
10. *Емеличев В., Мельников О., Сарванов В., Тышкевич Р.* Лекции по теории графов.
11. *Препарата Ф., Шеймос М.* Вычислительная геометрия. Введение.
12. *Романовский И.* Дискретный анализ.
13. *Шень А.* Программирование: теоремы и задачи.
14. *Липский В.* Комбинаторика для программистов.
15. *Menezes A., Van Oorschot P., Vanstone S.* Handbook of Applied Cryptography.

Ссылки в сети Интернет

16. http://olympiads.win.tue.nl/ioi – Официальный сайт международных олимпиад по информатике (IOI).
17. http://neerc.ifmo.ru/school – Сайт олимпиад школьников по информатике в Санкт-Петербурге и России. Содержит, в частности, материалы сборов.
18. http://www.informatics.ru – Сайт олимпиад школьников по информатике. Содержит, в частности, материалы сборов.
19. http://eccc.uni-tier.de/eccc-local/ECCC-LecutreNotes/IntroComplTh – Материалы лекций по теории сложности.
20. http://www.nada.kth.se/~johanh/ – Домашняя страница Джона Хестада, содержащая, в частности, курс лекций «Advanced Algorithms».

Бабенко Максим Александрович,
студент 4 курса кафедры математической логики и теории алгоритмов мехмата МГУ,
член жюри учебно-тренировочных сборов по информатике,
член научного комитета Всероссийской олимпиады школьников по информатике.

Станкевич Андрей Сергеевич,
студент 5 курса кафедры компьютерных технологий СПбГИТМО (ТУ),
член жюри учебно-тренировочных сборов по информатике,
председатель жюри Всероссийской командной олимпиады школьников по информатике.