Фамилия ____________________________ Имя ____________________________

1. Выберите все истинные утверждения относительно элементов двоичной кучи $h[0 \ldots n-1]$, в корне которой находится минимум (считайте, что i и j таковы, что все рассматриваемые элементы существуют).

□ $h[0] \le h[i]$	□ $h[0] \le h[n-1]$	□ $h[i] \le h[n-1]$
□ $h[i] \le h[i+1]$	□ $h[i] \le h[2i+1]$	□ $h[i] \le h[2i+2]$
□ $h[i] \le h[(i-1) \text{ div } 2]$	□ $h[(i-1) \text{ div } 2] \le h[i]$	□ $h[i] \ne h[j]$ при $i \ne j$

2. Условие оптимальности для префиксов в задаче о кратчайшем пути в ациклическом графе означает (выберите истинные утверждения):

   □ Префикс $s \rightsquigarrow v$ оптимального пути $s \rightsquigarrow v \rightsquigarrow u$ всегда является оптимальным путем от s до v

   □ Если префикс $s \rightsquigarrow v$ пути $s \rightsquigarrow v \rightsquigarrow u$ является оптимальным путем от s до v, то путь $s \rightsquigarrow v \rightsquigarrow u$ является оптимальным путем от s до u

   □ Если префикс $s \rightsquigarrow v$ пути $s \rightsquigarrow v \rightsquigarrow u$ является оптимальным путем от s до v, и фрагмент $v \rightsquigarrow u$ является оптимальным путем от v до u, то путь $s \rightsquigarrow v \rightsquigarrow u$ является оптимальным путем от s до u

3. Заполните таблицу параметров различных алгоритмов на графах. В графе SS отметьте, верно ли, что алгоритм предназначен для нахождения путей от одной вершины до всех, в графе W+ — требуется ли для корректности работы алгоритма неторицательность весов ребер, в графе T — оцените асимптотически время работы (V — число вершин в графе, E — число ребер).

Алгоритм	**SS**	**W+**	**T**
алгоритм Флойда			
алгоритм Форда-Беллмана			
алгоритм Дейкстры (простейший вариант)			
алгоритм Дейкстры (с двоичной кучей)			

4. 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

5. 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

   □ число не превыщающее кратчайшую длину пути, состоящего не более чем из k ребер, до вершины i

6. За какое время можно построить дерево отрезков для заданного массива?

   □ $O(n^2)$ □ $O(n \log n)$ □ $O(n)$ □ $O(\log n)$ □ $O(1)$

7. Выберите истинные утверждения про дерево Фенвика.

   □ Дерево Фенвика можно построить за $O(n)$.

   □ Построение дерева Фенвика требует увеличения размера массива до степени двойки.

   □ Дерево Фенвика требует $O(1)$ дополнительной памяти.

   □ Дерево Фенвика можно построить для операции «минимум».

   □ Дерево Фенвика можно построить для операции «xor».

   □ Дерево Фенвика можно построить для операции «умножение перестановок».

8. Выберите истинные утверждения про декартово дерево.
   - ☐ Декартово дерево для пар чисел (x_i, y_i) является двоичным деревом поиска по ключу x_i.
   - ☐ В декартовом дереве для пар чисел (x_i, y_i) соблюдается порядок кучи по ключу y_i.
   - ☐ В декартовом дереве для пар чисел (x_i, y_i) не может быть двух вершин с одинаковым ключом y_i.
   - ☐ Декартово дерево требует $O(1)$ дополнительной памяти.
   - ☐ Высота декартова дерева с n вершинами — $O(\log n)$.
   - ☐ Декартово дерево для n пар чисел (x_i, y_i) можно построить за $O(n)$.
   - ☐ Декартово дерево для n пар чисел (i, y_i) можно построить за $O(n)$.
   - ☐ Декартово дерево однозначно задается набором пар ключей (x_i, y_i).
   - ☐ Можно объединять два произвольных декартовых дерева за $O(\log n)$.
   - ☐ Можно объединять два произвольных декартовых дерева за время, пропорциональное сумме их размеров.
   - ☐ Можно делать операцию split по приоритетам за $O(\log n)$.
   - ☐ В декартовом дереве по неявному ключу вместо ключей используются приоритеты.

9. Постройте дерево отрезков для операции минимум и набора чисел: $\{21, 153, 7, 100, 9, 42, 256, 17, 194, 14\}$

10. Постройте дерево Фенвика для операции сложения и набора чисел: $\{21, 153, 7, 100, 9, 42, 256, 17, 194, 14\}$

11. Постройте skip-list для набора чисел: $\{21, 153, 7, 100, 9, 42, 256, 17, 194, 14\}$

12. За какое время возможно решение динамической задачи выбора минимума на отрезке (RMQ)? Выберите варианты, для которых вам известен алгоритм с временем предподготовки $P(n)$ и временем обработки запроса на изменение элемента или получения минимума на отрезке $Q(n)$. Для каждого из отмеченных вариантов укажите с помощью какой структуры данных эта оценка достигается и какое количество дополнительной памяти $M(n)$ — необходимо.

- ☐ $P(n) = O(1)$, $Q(n) = O(n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n)$, $Q(n) = O(\log n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n^2)$, $Q(n) = O(1)$ $M(n) =$ ________________
- ☐ $P(n) = O(n \log n)$, $Q(n) = O(\log n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n \log n)$, $Q(n) = O(1)$ $M(n) =$ ________________
- ☐ $P(n) = O(n)$, $Q(n) = O(1)$ $M(n) =$ ________________

13. За какое время возможно решение статической задачи выбора минимума на отрезке (RMQ)? Выберите варианты, для которых вам известен алгоритм с временем предподготовки $P(n)$ и временем обработки запроса получения минимума на отрезке $Q(n)$. Для каждого из отмеченных вариантов укажите с помощью какой структуры данных эта оценка достигается и какое количество дополнительной памяти $M(n)$ — необходимо.

- ☐ $P(n) = O(1)$, $Q(n) = O(n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n)$, $Q(n) = O(\log n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n^2)$, $Q(n) = O(1)$ $M(n) =$ ________________
- ☐ $P(n) = O(n \log n)$, $Q(n) = O(\log n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n \log n)$, $Q(n) = O(1)$ $M(n) =$ ________________
- ☐ $P(n) = O(n)$, $Q(n) = O(1)$ $M(n) =$ ________________

14. За какое время возможно выполнение m запросов о наименьшем общем предке (LCA) в дереве с n вершинами в режиме offline (Все запросы известны заранее)? Выберите варианты, для которых вам известен алгоритм с временем работы $P(n)$, укажите используемые структуры данных и алгоритм.

- ☐ $P(n) = O(nm)$ ________________
- ☐ $P(n) = O(m \log n)$ ________________
- ☐ $P(n) = O(m \log *n)$ ________________
- ☐ $P(n) = O(m + n)$ ________________
- ☐ $P(n) = O(1)$ ________________

15. За какое время возможно решение задачи нахождения наименьшего общего предка в дереве в режиме online (LCA)? Выберите варианты, для которых вам известен алгоритм с временем предподготовки $P(n)$ и временем обработки запроса $Q(n)$. Для каждого из отмеченных вариантов укажите необходимое количество дополнительной памяти $M(n)$ для известного вам метода.

- ☐ $P(n) = O(1)$, $Q(n) = O(n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n)$, $Q(n) = O(\log n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n^2)$, $Q(n) = O(1)$ $M(n) =$ ________________
- ☐ $P(n) = O(n \log n)$, $Q(n) = O(\log n)$ $M(n) =$ ________________
- ☐ $P(n) = O(n \log n)$, $Q(n) = O(1)$ $M(n) =$ ________________
- ☐ $P(n) = O(n)$, $Q(n) = O(1)$ $M(n) =$ ________________

16. За какое время возможно реализации структуры данных для системы непересекающихся множеств? Выберите варианты, для которых вам известен алгоритм с указанным временем работы для m операций `get` и n операций `union` ($m \geq n$), и укажите используемые структуры данных.

- ☐ $O(m+n^2)$
- ☐ $O(m+n\log n)$
- ☐ $O(m\log n)$
- ☐ $O(m\log^* n)$
- ☐ $O(m+n\log^* n)$
- ☐ $O(m+n)$

17. Выберите истинные утверждения.

- ☐ Если объекты равны, то хеши могут быть равны.
- ☐ Если объекты равны, то хеши могут быть не равны.
- ☐ Если хеши равны, то объекты могут быть равны.
- ☐ Если хеши равны, то объекты могут быть не равны.
- ☐ Если объекты не равны, то хеши всегда не равны.
- ☐ Если хеши не равны, то объекты всегда не равны.
- ☐ $h(x) = 1$ является корректной хеш-функцией.
- ☐ $h(x) = \mathrm{rand}\{0,1\}$ является корректной хеш-функцией.

18. Для хеширования строки s, состоящей из маленьких латинских букв, была применена формула $\sum\limits_{i=1}^{n}(s_i - \texttt{'a'})p^{n-i}$. Приведите пример двух различных строк, имеющих одинаковый хеш.

19. Выберите истинные утверждения.

- ☐ Алгоритм Ахо-Корасик позволяет найти количество вхождений строк S_i в текст T за время $O(\sum|S_i| + |T|)$.
- ☐ Алгоритм Ахо-Корасик позволяет найти все вхождения строк S_i в текст T за время $O(\sum|S_i| + |T|)$.
- ☐ Алгоритм Ахо-Корасик позволяет найти наибольшую общую подстроку двух строк с длинами n и m за $O(n+m)$.
- ☐ Алгоритм Ахо-Корасик позволяет найти наибольший префикс строки S входящий в строку T за $O(|S| + |T|)$.
- ☐ Глубина вершины в которую указывает суффиксная ссылка, построенная алгоритмом Ахо-Корасик для одной строки, совпадают с префикс функцией для этой строки.
- ☐ Суффиксные ссылки, построенные алгоритмом Ахо-Корасик для одной строки, совпадают с Z-функцией для этой строки.

20. Пусть $Z[i]$ - Z-функция, а π - префикс функция. Выберите истинные утверждения.

- ☐ Для строки S и любого $i < j$ верно, что $Z[i] \leq Z[j]$.
- ☐ Для строки S и любого $i < j$ верно, что $Z[i] \geq Z[j]$.
- ☐ Для строки S и любого $i < j$ верно, что $\pi[i] \leq \pi[j]$.
- ☐ Для строки S и любого $i < j$ верно, что $\pi[i] \geq \pi[j]$.
- ☐ Для строки S и любого i верно, что $\pi[i] \leq Z[i]$.
- ☐ Для строки S и любого i верно, что $\pi[i] \geq Z[i]$.
- ☐ Для строки S и любого i верно, что $Z[i] \leq i$.
- ☐ Для строки S и любого i верно, что $Z[i] \geq i$.
- ☐ Для строки S и любого i верно, что $\pi[i] \leq i$.
- ☐ Для строки S и любого i верно, что $\pi[i] \geq i$.
- ☐ Для строки S и любого i верно, что $Z[i] \leq Z[Z[i]]$.
- ☐ Для строки S и любого i верно, что $Z[i] \geq Z[Z[i]]$.
- ☐ Для строки S и любого i верно, что $Z[i] \leq Z[\pi[i]]$.
- ☐ Для строки S и любого i верно, что $Z[i] \geq Z[\pi[i]]$.
- ☐ Для строки S и любого i верно, что $\pi[i] \leq \pi[\pi[i]]$.
- ☐ Для строки S и любого i верно, что $\pi[i] \geq \pi[\pi[i]]$.
- ☐ Z-функция позволяет найти наибольшую общую подстроку двух строк S и T за $O(|S| + |T|)$.
- ☐ Z-функция позволяет найти все вхождения строки S в текст T за $O(|S| + |T|)$.
- ☐ Z-функцию для строки S можно вычислить за $O(|S|)$.

21. Подстроку длины m можно найти в строке длины n за время:

☐ $O(m^2n)$ ☐ $O(mn^2)$ ☐ $O(nn)$ ☐ $O(m+n)$ ☐ $O(n/m)$

22. Выберите верные утверждения:
    - ☐ Суффиксный массив (СМ) содержит номера суффиксов строки в порядке увеличения длины.
    - ☐ СМ содержит номера суффиксов строки в лексикографическом порядке.
    - ☐ СМ содержит номера префиксов строки в порядке увеличения длины.
    - ☐ СМ содержит номера префиксов строки в лексикографическом порядке.
    - ☐ СМ строки длины n можно построить за время $O(n^2)$.
    - ☐ СМ строки длины n можно построить за время $O(n \log n)$.
    - ☐ СМ строки длины n можно построить за время $O(n)$.
    - ☐ СМ строки длины n позволяет искать в ней подстроку длины m за $O(nm)$.
    - ☐ СМ строки длины n позволяет искать в ней подстроку длины m за $O(n \log m)$.
    - ☐ СМ строки длины n позволяет искать в ней подстроку длины m за $O(m \log n)$.
    - ☐ СМ строки длины n позволяет искать в ней подстроку длины m за $O(m + \log n)$.
    - ☐ СМ строки длины n позволяет искать в ней подстроку длины m за $O(n)$.
    - ☐ СМ строки длины n позволяет искать в ней подстроку длины m за $O(m)$.

23. Выберите верные утверждения:
    - ☐ Суффиксный автомат (СА) допускает все суффиксы заданной строки.
    - ☐ СА допускает все префиксы заданной строки.
    - ☐ СА допускает все подстроки заданной строки.
    - ☐ СА для строки длины n содержит $O(n)$ состояний.
    - ☐ СА для строки длины n содержит $O(n)$ переходов.
    - ☐ СА для строки длины n можно построить за время $O(n^2)$.
    - ☐ СА для строки длины n можно построить за время $O(n \log n)$.
    - ☐ СА для строки длины n можно построить за время $O(n)$.
    - ☐ СА для строки длины n позволяет искать в ней подстроку длины m за $O(nm)$.
    - ☐ СА для строки длины n позволяет искать в ней подстроку длины m за $O(n \log m)$.
    - ☐ СА для строки длины n позволяет искать в ней подстроку длины m за $O(m \log n)$.
    - ☐ СА для строки длины n позволяет искать в ней подстроку длины m за $O(m + \log n)$.
    - ☐ СА для строки длины n позволяет искать в ней подстроку длины m за $O(n)$.
    - ☐ СА для строки длины n позволяет искать в ней подстроку длины m за $O(m)$.

24. Выберите верные утверждения:
    - ☐ Суффиксное дерево содержит все суффиксы заданной строки.
    - ☐ Суффиксное дерево содержит все префиксы заданной строки.
    - ☐ Суффиксное дерево содержит все подстроки заданной строки.
    - ☐ Сжатое суффиксное дерево строки длины n содержит $O(n)$ вершин.
    - ☐ Сжатое суффиксное дерево строки длины n содержит $O(n)$ ребер.
    - ☐ Сумма длин пометок ребер суффиксного дерева строки длины n есть $O(n)$.
    - ☐ Сумма длин пометок ребер суффиксного дерева строки длины n есть $O(n^2)$.
    - ☐ Сжатое суффиксное дерево строки длины n можно построить за время $O(n^2)$.
    - ☐ Сжатое суффиксное дерево строки длины n можно построить за время $O(n \log n)$.
    - ☐ Сжатое суффиксное дерево строки длины n можно построить за время $O(n)$.
    - ☐ Сжатое суффиксное дерево строки длины n позволяет искать в ней подстроку длины m за время $O(nm)$.
    - ☐ Сжатое суффиксное дерево строки длины n позволяет искать в ней подстроку длины m за время $O(n \log m)$.
    - ☐ Сжатое суффиксное дерево строки длины n позволяет искать в ней подстроку длины m за время $O(m \log n)$.
    - ☐ Сжатое суффиксное дерево строки длины n позволяет искать в ней подстроку длины m за время $O(m + \log n)$.
    - ☐ Сжатое суффиксное дерево строки длины n позволяет искать в ней подстроку длины m за время $O(n)$.
    - ☐ Сжатое суффиксное дерево строки длины n позволяет искать в ней подстроку длины m за время $O(m)$.

25. Постройте суффиксный массив для строки: «ababaababba».

26. Постройте суффиксный автомат и дерево правых контекстов для строки: «ababaababba».

27. Постройте сжатое суффиксное дерево для строки: «ababaababba».

28. Укажите, для каких из приведенных задач вы знаете алгоритм решения за полиномиальное время. Укажите его асимптотику.

Задача	**Время работы алгоритма**
Поиск максимального паросочетания в двудольном графе	
Поиск максимального паросочетания в произвольном графе	
Поиск паросочетания минимального веса в двудольном графе (задача о назначениях)	
Поиск паросочетания минимального веса в произвольном графе	
Поиск минимального вершинного покрытия в двудольном графе	
Поиск минимального вершинного покрытия в произвольном графе	
Поиск максимального потока	
Поиск потока минимальной стоимости	

29. Выберите истинные утверждения.

- ☐ Минимальный поток из s в t в сети G равен максимальному s-t разрезу.
- ☐ В сети G существует единственный максимальный поток тогда и только тогда, когда в ней существует единственный минимальный разрез.
- ☐ Алгоритм Форда-Фалкерсона работает за полиномиальное время от количества ребер.
- ☐ Максимальный поток в сети с целыми пропускными способностями всегда целый.
- ☐ Любой поток можно разложить в сумму потоков вдоль путей из истока в сток.
- ☐ Любой поток из s в t в сети G не больше любого s-t разреза в сети G.
- ☐ Поток является максимальным тогда и только тогда когда в остаточной сети нет циклов.
- ☐ Поток имеет величину 0 тогда и только тогда, когда поток по каждому ребру равен 0.
- ☐ Поток по ребру равен потоку по обратному ребру.
- ☐ Поток f является максимальным, если на любом пути из s в t в G найдется насыщенное ребро.
- ☐ Поток f является максимальным, если на любом пути из s в t в G_f найдется насыщенное ребро.

30. Выберите верные утверждения:

- ☐ При анализе игры на ациклическом графе используется поиск в глубину.
- ☐ При анализе игры на ациклическом графе используется поиск в ширину.
- ☐ При анализе игры на графе с циклами используется поиск в глубину.
- ☐ При анализе игры на графе с циклами используется поиск в ширину.
- ☐ Все позиции в игре на ациклическом графе являются либо выигрышными, либо проигрышными.
- ☐ Все позиции в игре на графе с циклами являются либо выигрышными, либо проигрышными.
- ☐ Если позиция в игре на графе является ничейной то она лежит на цикле.
- ☐ Если позиция в игре на графе лежит на цикле, то она является ничейной.
- ☐ Позиции в игре на графе является ничейной тогда и только тогда, когда она лежит на цикле.

31. Выберите верные утверждения:

- ☐ Сумма игр на графах, содержащих m и n позиций, соответственно, является игрой, содержащей $m + n$ позиций.
- ☐ Сумма игр на графах, содержащих m и n позиций, соответственно, является игрой, содержащей mn позиций.
- ☐ Две игры на графе являются эквивалентными по Гранди, если они имеют одинаковый исход.
- ☐ Две игры на графе являются эквивалентными по Гранди, если они имеют одинаковый исход при суммировании с любой игрой.
- ☐ Две игры на графе являются эквивалентными по Гранди, если они имеют одинаковый исход при суммировании с ничейной игрой.
- ☐ Любая игра на ациклическом графе эквивалентна игре ним.
- ☐ Любая игра на графе эквивалентна игре ним.
- ☐ Любая игра на графе, которая не является ничейной, эквивалентна игре ним.
- ☐ Функция Гранди позиции u в игре на графе равна сумме функций Гранди позиций, в которые из нее возможен ход.
- ☐ Функция Гранди позиции u в игре на графе равна минимуму из функций Гранди позиций, в которые из нее возможен ход.
- ☐ Функция Гранди позиции u в игре на графе равна максимуму из функций Гранди позиций, в которые из нее возможен ход.
- ☐ Функция Гранди позиции u в игре на графе равна максимуму из функций Гранди позиций, в которые из нее возможен ход, плюс один.
- ☐ Функция Гранди позиции u в игре на графе равна минимальному целому неотрицательному числу, не встречающемуся среди функций Гранди позиций, в которые из нее есть ход.

32. При суммировании игр с функциями Гранди a и b получается игра с функцией Гранди:

    □ $a + b$ □ ab □ $a \oplus b$ □ $\min(a, b)$ □ $\max(a, b)$ □ $\max(a, b) + 1$

33. Приведите пример, когда алгоритм Мелькмана не находит корректную выпуклую оболочку.

34. Выберите истинные утверждения:

    - □ Можно проверить непустоту пересечения многоугольника с прямой за $O(\log n)$.
    - □ Можно проверить непустоту пересечения выпуклого многоугольника с прямой за $O(\log n)$.
    - □ Можно проверить непустоту пересечения многоугольника с прямой и, если оно непусто, найти точку пересечения за $O(\log n)$.
    - □ Можно проверить непустоту пересечения выпуклого многоугольника с прямой и, если оно непусто, найти точку пересечения за $O(\log n)$.
    - □ Можно проверить непустоту пересечения многоугольника с прямой за $O(n)$.
    - □ Можно проверить непустоту пересечения выпуклого многоугольника с прямой за $O(n)$.
    - □ Можно проверить непустоту пересечения многоугольника с прямой и, если оно непусто, найти точку пересечения за $O(n)$.
    - □ Можно проверить непустоту пересечения выпуклого многоугольника с прямой и, если оно непусто, найти точку пересечения за $O(n)$.
    - □ Можно проверить непустоту пересечения n полуплоскостей за $O(n^3)$.
    - □ Можно проверить непустоту пересечения n полуплоскостей за $O(n^2 \log n)$.
    - □ Можно проверить непустоту пересечения n полуплоскостей за $O(n^2)$.
    - □ Можно проверить непустоту пересечения n полуплоскостей за $O(n \log n)$.
    - □ Можно проверить непустоту пересечения n полуплоскостей за $O(n)$.