Вариант 1

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^3 \log^{10} n$; b) $n^{\sqrt{n}}/n^{15}$; c) $n^{3.01}/\log^{100} n$; d) $n^{7/10}/\log^4 n$; e) $2^{n\log(n)}$;
f) $n3^n$; g) $n^{0.15}\log^5 n$; h) $n^{\log^2 n \log\log n}$; i) $n^2 \log^n n$; j) $n^{0.5}\log\log n$.

2 .(15) Пусть в нагруженном дереве вершинам приписаны имена. Написать нерекурсивный алгоритм, печатающий эти имена в префиксном порядке.

3. (10) Пусть A - матрица размера $n \times n$. Оцените сложность следующего алгоритма.
procedure PROC1 (A):
{ FOR i = 1 TO n DO
FOR j = i TO n DO
$B[i,j] := max\{A[k,j] | i-1 < k < n+1\}$
}

4. (10) Время работы алгоритма A описывается соотношением $T(n) = 7T(n/2) + n^2$, а время работы алгоритма B - соотношением $T(n) = aT(n/4) + n^2$.

При каком наибольшем целом a алгоритм B асимптотически быстрее алгоритма A.

5. (20) *Проблема уникальности элементов* состоит в определении по последовательности из n элементов $a_1, a_2, \ldots, a_n$ того, что $a_i \neq a_j$ при $i \neq j$.

Докажите, что всякое дерево сравнений, решающее проблему уникальности элементов для всех последовательностей длины n, имеет сложность не менее $\log_2(n!)$.

6. (20) Пусть массив $A[1..n]$ содержит все целые числа от 0 до n, кроме одного. Предположим, что за одно действие можно просмотреть заданный бит заданного элемента A. Предложите алгоритм сложности $O(n)$ для нахождения недостающего числа.

7 (10) Пусть исходный массив A уже отсортирован в порядке возрастания. Каким будет время его сортировки с помощью дерева? А каким оно будет, если он отсортирован в порядке убывания?

8.(15) Используя алгоритм Янгера-Касами, проверить входит ли слово *aaabbcca* в кс-язык, порожденный следующей кс-грамматикой ($N = \{A, B, C, S\}$, $T = \{a, b, c\}$:
$S => AA|AB$
$A => a|AB|CA$
$B => b|BB|BC$
$C => c|CC|CA$

Вариант 2

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $2n^{\log n}\log^7 n$; b) $3n^{\sqrt{\log n}}$; c) $n^2/log^{100}n$; d) $n^{199/100}$ e) $2^{n\log n}/n^2$; f) n^{2n};
g) $n^{2.6}/\log^{10} n$; h) $n\log n\log\log n$; i) $n\log^4 n$; j) $n^{2.5}\log\log^5 n$.

2. (15) Написать битовую линейную программу в базисе $\{\wedge, +(mod 2)\}$ для вычисления квадрата трехразрядного двоичного числа и оценить ее сложность.

3. (15). Оцените количество сравнений, требующихся следующей процедуре нахождения наибольшего и наименьшего элементов множества.
Вход: мн-во S из n эл-ов, n – степень двойки.
Выход: (max, min) - максимальный и минимальный эл-ты S.
procedure MAX-MIN(S):
if $|S| = 2$ **then** { пусть $S = \{a, b\}$; выдать$(MAX(a, b), MIN(a, b))$ }
else { разбить S на два равных подмн-ва S_1 и S_2;
$(max1, min1) := MAX - MIN(S1)$;
$(max2, min2) := MAX - MIN(S2)$;
выдать $(MAX(max1, max2), MIN(min1, min2))$

4.(15) Выбрать представление дерева и написать алгоритм подсчета для каждой вершины числа ее потомков. Какова его сложность?

5. (20) Пусть структура ООО "Рога и копыта"задана деревом подчиненности с директором в корне дерева, в котором каждому сотруднику (вершине) приписан неотрицательный рейтинг. Директор приказал своему заместителю по кадрам пригласить на праздник "День животновода"часть сотрудников так, чтобы никто из приглашенных не оказался на празднике вместе со своим непосредственным начальником. Используя динамическое программирование, разработайте алгоритм, составляющий список приглашенных сотрудников с максимальным суммарным рейтингом. Оцените его сложность.

6.(15) Модифицировать алгоритм МИНК так, чтобы первоначальная последовательность разбивалась на отрезки длины 7 (а не 5). Оценить сложность получившегося алгоритма.

7 .(10) Упорядочить последовательность $3, 16, 23, 4, 11, 13, 8, 2, 11, 9$ применяя алгоритм сортировки деревом. Сколько перемещений элементов производится в процессе сортировки?

8.(10) Используя алгоритм лексикографической сортировки, упорядочить последовательность:
ОЛВК,ЛООВ, ОКОВ,ВОЛО,ЛОКК,КВОК,КОЛО.

Вариант 3

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^3/\log^5 n$; b) $n^{\sqrt{n}}/n^3$; c) $n^{1.9}\log^{100} n$; d) $n^{0,1}/\log^{14} n$; e) $n^{n/\log n}$;

f) $n!/3^n$; g) $n^{0.5}\log^{10} n$; h) $n^{2\log n+\log\log n}$; i) $n^3\log^n n$; j) $n^3(\log\log n)^{10}/\log n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$: следующей процедуры. (n — степень тройки).

procedure PEL(A[m..n]):

{
$p := [(n-m+1)/3]; q := [(n-m+1)/2];$
$y :=$ МИНК$(6, A[m..n]); A[n/3-1] := 2 * y;$
$PEL(A[m..m+p-1]); PEL(A[m+p..m+2p-1]); PEL(A[m+2p..n]);$
$PEL(A[m+q..m+q+p-1])$
}

3.(15) Выбрать представление дерева и написать алгоритм подсчета для каждой вершины ее глубины. Какова его сложность?

4.(10) Какие ответы будет давать "стратегия дьявола"из теоремы Кислицына в ответ на следующие вопросы:
$a > b?; a > c?; c > b?; e > d?; d > f; a > e?; f > e?; f > g?; f > c?; a > f?; g > e?; g > a?$
(если ответ не определен, пусть "побеждает"игрок, стоящий в сравнении слева). Какое отношение "X превосходит Y"получается в результате?

5.(20) Пусть A - упорядоченный массив положительных и отрицательных целых чисел: $A[1] < A[2] < ... < A[n]$. Написать алгоритм нахождения такого i, что $A[i] = i$ (если такое i существует). Оценить его сложность ("хороший"алгоритм имеет сложность $O(\log n)$).

6.(15) Пусть в массиве $A[1..N]$ первые n ($n < N$) элементов представляют сортирующее дерево. Опишите алгоритм добавления к этому дереву нового элемента $A[n+1]$ и оцените его сложность.

7.(15) Доказать правильность алгоритма КОД, строящего оптимальное дерево двоичного поиска (т.е. показать индукцией по l правильность определения стоимостей $c_{i,j}$ и корней $r_{i,j}$).

8.(10) Найти медиану следующей последовательности с помощью алгоритма линейной сложности:

7, 2 ,5, 81,15, 4, 3, 7, 2, 1,17, 8, 9, 5, 26, 23, 1, 13, -2, 0, 41, 15, 20, 17, 5, 12, 3.

Вариант 4

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $(n!)\log^5 n$; b) $2^{\sqrt{n}} n^2$; c) $n/\log^{100} n$; d) $n^{19/20}\log^7 n$; e) $2^{n!}/n^n$;

f) $n^4 3^n$; g) $n^{2.05}/\log^{10}(n+1)$; h) $n^{\log n}$; i) $n^2\log^n n$; j) $n^{0.99}/\log\log^5 n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$ следующей процедуры (n – степень 4.)
procedure$PEL(A[m..n])$:
$\{$
$p := [(n-m+1)/4]; q := [(n-m+1)/2];$
$MAKE(A[m..n]);$
$PEL(A[m..m+p-1]); PEL(A[m+p..m+2p-1]); PEL(A[m+2p..n]);$
$PEL(A[m+q..m+q+p-1])$
$\}$

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $O((n-m)^2)$.

3.(15) Предложите алгоритм сложности $O(n)$, который проверяет, содержит ли ориентированный граф, заданный матрицей смежностей, вершину, к которой ведут дуги из остальных $(n-1)$ вершин и из которой не выходит ни одна дуга.

4.(15) Доказать, что алгоритм, использующий метод динамического программирования для распознавания кс-языка, порождаемого кс-грамматикой в нормальной форме Хомского, имеет сложность $O(n^3)$.

5.(20) Дано n целых чисел от 1 до k. Предложите алгоритм, который подвергает их предварительной обработке, а затем за время $O(1)$ отвечает на любой вопрос вида "сколько чисел из данного набора лежит между a и b?"Время предварительной обработки должно быть $O(n+k)$.

6.(10) Найти оптимальное дерево двоичного поиска для элементов a, b, c, d, e с вероятностями $p_i = 0.1; 0.3; 0.05; 0.1; 0.2$ и $q_i = 0.1; 0; 0.05; 0; 0; 0.1$.

7.(15) Построить РАМ-программу, вычисляющую по данному n $f(n) = 2^{2^n}$ за $O(n)$ шагов. Чему равна сложность этого вычисления при логарифмическом весе команд.

8.(10) Построить алгоритмом ПОСОРТДЕР сортирующее дерево для следующего массива: $A = (15, 3, 17, 10, 43, 19, 6, 22, 9)$. Сколько операций сравнения при этом выполняется?

Вариант 5

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):

a) $n\log^3(n!)$; b) $2^{\sqrt{n}}n$; c) $n/\log^{10} n$; d) $n^{9/10}\log^4 n$; e) $2^{n!/n^n}$;

f) $25n$; g) $n^{1.05}/\log^{40}(n+10)$; h) $n\log n$; i) $n^2(\log n)^n$; j) $n^{0.91}\log^7 n$.

2.(15) Выбрать представление дерева и написать алгоритм подсчета для каждой вершины ее высоты. Какова его сложность?

3.(15) Дополните алгоритм вычисления минимального числа операций при умножении матриц, чтобы он выдавал и оптимальный порядок умножений (соответствующую расстановку скобок).

4. (15) Разработайте алгоритм, который по заданному числу k находит в данном множестве S мощности n его k элементов, ближе всего расположенных к медиане. Сложность алгоритма должна быть $O(n)$.

5.(15) Используя алгоритм Янгера-Касами, проверить входит ли слово $aabbbca$ в кс-язык L_G, порожденный следующей кс-грамматикой $G = (\{S, A, B, C\}, \{a, b, c\}, P, S)$:
$P :$
$S => AA|AB$
$A => a|AA|CA$
$B => b|BB|BC$
$C => c|CC|CA$
Если оно входит в L_G, то построить дерево его вывода.

6. (20) Доказать, что среднее число сравнений для успешного поиска элемента в упорядоченном массиве $A[1..n]$ (n - степень 2) методом бинарного поиска (все элементы A считаются равновероятными) равно $O(logn)$.

7. (10). Используя алгоритм лексикографической сортировки, упорядочить следующую последовательность слов длины 4:
ВОВА, ВЛАА, ОПАЛ, ЛОЛА, ОВАЛ, АОПП, ПАПА, ЛАВА, ПОЛО.

8. (10). Найти медиану следующей последовательности A с помощью алгоритма МИНК линейной сложности:
$A = \ 18, 3, 5, 1, 12, 4, 3, 7, 12, 1, 4, 5, 22, 8, 9, 9, 59, 13, 13, -2, 0, 8, 15, 23, 97, 5, 34.$

Вариант 6

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):

a) $n^2 \log^3 n$; b) $n^{\sqrt{n}}/n^3$; c) $n^3/\log^{100} n$; d) $n^{21/10}/\log^4 n$; e) $2^{n \log n}$;

f) n^{3n}; g) $n^{0.5} \log^{10} n$; h) $n \log n \log\log n$; i) $n^2 \log\log n$; j) $n^{0.51} \log\log n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$: следующей процедуры (n - степень тройки).
procedure$PEL(A[m..n])$:
{
$p := [(n-m+1)/3]; q := [(n-m+1)/2];$
$MAKE(A[m..n]);$
$PEL(A[m..m+p-1]); PEL(A[m+p..m+2p-1]);$
}

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $\theta((n-m)^2)$.

3. (15) Каково максимальное число вершин высоты h в сортирующем дереве из n элементов? Ответ обоснуйте.

4. (20) . Пусть имеется n достоинств монет, $T[1..n]$ - массив, задающий их значения. Для $1 \leq i \leq n$ и $1 \leq j \leq L$ пусть $C(i,j)$ обозначает минимальное число монет достоинств $T(1), T(2), ..., T(i)$, которыми можно разменять сумму j. Определить $C(i,j)$ рекурсивно и построить алгоритм, вычисляющий $C(n, L)$ и способ наилучшего размена L.

5. (15) Пусть дан массив $A[1..n]$ целых чисел и массив $P[1..n]$, задающий перестановку чисел $1, 2, ..., n$. Написать алгоритм, располагающий элементы массива A в порядке, задаваемом P: на выходе $A'[i] = A[P[i]]$, который работал бы внутри A. Каково время работы алгоритма в худшем случае ?

6. (15) Построить линейную битовую программу в базисе $\{\neg, \wedge, \vee\}$, вычисляющую сумму двух n-разрядных двоичных чисел за время $O(n)$.

7.(10) Найти оптимальное дерево двоичного поиска для элементов a, b, c, d, e с вероятностями $P_i = 0.2; 0.25; 0.05; 0.2; 0.05$ и $q_i = 0; 0; 0.05; 0; 0.1; 0.1$.

8.(10) Найти 5-ый наименьший элемент следующей последовательности с помощью алгоритма МИНСК (с линейной сложностью в среднем):
7, 2, 6, 1,10, 4, 1, 23, 4,12, 8, 9, 56, 23,13, -2, 0,4, 15, 20, 17, 4.
(во 2-ой строке выбирать медиану первого, последнего и среднего элементов).

Вариант 7

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):

a) $2^{n\log n}$; b) $n^{\sqrt{\log n}}$; c) $n^2/\log 1000n$; d) $n^{19/10}$; e) $2^n \log n/n$;

f) n^{2^n}; g) $n^{1.5}\log^{10} n$; h) $n\log n\log\log n$; i) $n\log^4 n$; j) $n^{2.5}/\log\log^7 n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$ следующей процедуры (n– степень 5)
. **procedure**$PEL(A[m..n])$:
begin
$p := [(n-m+1)/5]; q := [(n-m+1)/2];$
$MAKE(A[m..n]);$
$PEL(A[m..m+p-1]); PEL(A[m+p..m+2p-1]); PEL(A[m+q..m+q+p-1])$
end

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $(n-m)log(n-m)$.

3.(15) Докажите, что в любом неориентированном связном графе $G=(V,E)$ выполнено неравенство $|E| >= |V|-1$, а если G – дерево, то $|E| = |V|-1$.

Предложите на основании этого равенства алгоритм, проверяющий по матрице смежности связного графа, является ли он деревом. Какова его сложность?

4.(20) Написать эффективный алгоритм для определения порядка вычисления произведения матриц $M_1 \times M_2 \times \ldots \times M_n$, минимизирующего число умножений элементов, в случае, когда каждая матрица M_i имеет один из размеров $1\times 1, 1\times d, d\times 1, d\times d$ при некотором фиксированном d . Он должен быть более быстрым, чем алгоритм, который оптимизирует умножение произвольных матриц.

5.(20) Пусть $S_1, S_2, ..., S_k$ - множества чисел, лежащих между 1 и n, и их суммарная мощность не больше n. Описать алгоритм, упорядочивающий все множества S_i за время $O(n)$.

6.(10) Пусть исходный массив A уже отсортирован в порядке убывания. Каким будет время его сортировки с помощью алгоритма быстрой сортировки?

7. (10) Построить для дерева , заданного списком дуг, представляющее его бинарное дерево $_1$. Описать T_1 с помощью матрицы смежности.
$T = \{(v1,v2),(v1,v3),(v2,v3),(v2,v4),(v2,v5),(v2,v6),(v3,v7),(v3,v8),(v3,v9),(v3,v10),(v5,v11)\}$.

8 (10) . Начертить двоичное дерево, соответствующее формуле
$((a+b)*(w-s+c)+x)/(d-(f+g*(z+x+v)))$.
Выписать его вершины в суффиксном и префиксном порядках.

Вариант 8

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^2/\log^3 n$; b) $n^{\sqrt{n}}/n^3$; c) $n^{1.99}\log^{100} n$; d) $n^{0.1}\log^4 n$; e) $n^{n\log n}$;

f) $n!/3^n$; g) $n^{0.05}\log^{10} n$; h) $n^2 \log n \log\log^9 n$; i) $n^2\log^n n$; j) $n^2(\log\log n)^{10}/logn$.

2.(15) Оценить сложность вызова $PER(A[1..N])$ следующего алгоритма (N - степень тройки).
procedure$PER(A[m..n])$:
begin
$p := n - m + 1;$
if $p \leq 2$ **then return**(0)
else begin
ПУЗЫРЕК($[m..n]$);
$PER(A[m..m + p/3]); PER(A[m + 2p/3..n - 1])$
end end

Здесь ПУЗЫРЕК - это вызов процедуры пузырьковой сортировки.

3.(10) Докажите, что в любом бинарном дереве число вершин степени 2 на единицу меньше числа листьев.

4. (20) На квадратной доске A размера $N \times N$ расставлены целые неотрицательные числа $A[i, j] \geq 0$. Черепашка, находящаяся в левом верхнем углу, мечтает попасть в правый нижний. При этом она может переползать только в клетку справа или снизу и хочет, чтобы сумма всех чисел, оказавшихся у нее на пути, была бы максимальной. Предложите алгоритм для определения этой суммы и соответствующего ей пути черепашки. Оцените его сложность.

5. (20) Пусть массив $A[1..n]$ содержит все целые числа от 0 до n, кроме одного. Предположим, что за одно действие можно просмотреть заданный бит заданного элемента A. Предложите алгоритм сложности $O(n)$ для нахождения недостающего числа.

6.(15) Доказать, что время работы процедуры ПЕРЕСЫПКА(1,i) для вершины i высоты k не больше $O(k)$.

7.(10) Отсортировать следующую последовательность методом слияния.
7, 2, 5, 81,10, 4, 3,7,2, 1, 23, 5.
Определить количество вызовов процедуры СЛИ(A, B).

8.(10) Найти 5-ый минимальный элемент следующей последовательности с помощью алгоритма МИНК линейной сложности:
17, 2, 5, 81,10,14,3,17,2,1,3,41,5,8,5,56,23,13,-2,0,4,32,20,17.

Вариант 9

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):

a) $n^4 \log^6 n$; b) $2^{\sqrt{n}}/n^3$; c) $n^5/\log^{100} n$; d) $2^{n \log n}$; e) $n^{41/10}/\log^8 n$;

f) $n^{\log n!}$; g) $n^{0.5} \log^{10} n$; h) $n \log n \log \log n$; i) $n^{2^{\log \log n}}$; j) $n^{4.1} \log \log^5 n$.

2.(15) Оценить сложность вызова $PR(a[1..N])$ для следующего алгоритма (N - степень тройки).
procedure$PR(A[m..n])$:
begin
if$(n - m \leq 1)$ **then return**(1)
else begin
$p := n - m + 1$;
обработать$(A[1..p])$; обработать$(A[m + 2p..n])$;
$PR(A[m..m + p/3 - 1]); PR(A[m + 2p/3..n])$
end
end
(сложность процедуры "обработать$(x, A[m, n])$"считать равной $O(1)$).

3. (15) Напишите нерекурсивный вариант процедуры суффиксного обхода бинарного дерева. Докажите его корректность и оцените сложность.

4. (15) Модифицировать алгоритм быстрой сортировки так, чтобы он работал за время $O(n \log n)$ в худшем случае. (*Указание:* используйте идею алгоритма МИНК).

5. (20) Абзац текста состоит из n слов длиной $l_1, l_2, ..., l_n$. Используя динамическое программирование, предложить алгоритм размещения слов абзаца в строках длины m с минимальным числом "лишних"пробелов. Число "лишних"пробелов в строке равно $m-$(суммарная длина слов в этой строке) - (число слов в строке) $+1$). Число "лишних"пробелов в абзаце равно сумме кубов "лишних"пробелов в строках.

6. (15) Найдите оптимальное глобальное выравнивание для слов "МУХОМОР"и "МРУМУР". Пусть функция веса сравнения двух символов $t(x, y)$ задана следующими равенствами: $t(x, x) = 1, t(-, x) = t(x, -) = -1, t(x, y) = -2(x \neq y)$, $x, y \in \{$М, О, Р, У, Х$\}$.

7.(10) Упорядочить последовательность 2,16,23,14,11,3,8,2,11, 6, применяя алгоритм сортировки деревом. Сколько сравнений производится в процессе сортировки?

8.(10) Найти 6-ой минимальный элемент следующей последовательности с помощью алгоритма линейной сложности:
7,2,5,81,10,14,3,17,2,1,3,4,5,8,25,56,23,13,-2,10,4,32,20,17.

Вариант 10

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $\log(n!)\log^5 n$; b) $n^{\sqrt{n}}n^2$; c) $n/\log^{100} n$; d) $n^{21/20}\log^7 n$; e) $2^{n!}/n^n$;

f) 3^n; g) $n^{2.05}/\log^{10}(n+1)$; h) $n^{\log n}$; i) $n^2\log^n n$; j) $n^{0.99}/\log\log^5 n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$: следующей процедуры (n - степень 4).
procedure$PEL(A[..n])$:
begin
$p := (n-m+1)/4; q := (n-m+1)/2;$
$MAKE(A[m..n]);$
$PEL(A[m..m+p-1]); PEL(A[m+2p..n]);$
$PEL(A[m+q..m+q+p-1])$
end

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $(n-m)log(m-n)$.

3.(15) Докажите, что неориентированный граф $G=(V,E)$ является деревом $\leftrightarrow$ G - ациклический и $|E|=|V|-1$. Предложите на основании этого равенства алгоритм, проверяющий по матрице смежности ациклического графа, является ли он деревом. Какова его сложность?

4.(20) Пусть правила умножения элементов алфавита $\{a,b,c\}$ задаются равенствами: $aa=b, ab=c, ac=a, ba=a, bb=b, bc=c, ca=a, cb=a, cc=c$. Написать алгоритм, проверяющий по слову w в алфавите $\{a,b,c\}$, можно ли его сократить до a. Например, для $w=bbbba$ имеем $(b(bb))(ba)=a$. Оценить сложность предложенного алгоритма.

5.(15) Докажите, что сортирующее дерево из n элементов содержит не более $[n/2(h+1)]$ вершин высоты h.

6.(15) Как изменить алгоритм сортировки методом слияния, чтобы он учитывал наличие в массиве упорядоченных отрезков. Какова будет его сложность, если число таких отрезков $\leq k$?

7. (10) Построить линейную программу, вычисляющую определитель 3 x 3 матрицы по ее элементам. Оценить время ее работы и используемую память.

8. (10) Упорядочить последовательность 23,16,23,14,11,3,8,32,11, 8, применяя алгоритм быстрой сортировки (разбиение массива производить относительно среднего арифметического его концов). Сколько сравнений производится ?

Вариант 11

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $2^{n/\log n}$; b) $n^{\sqrt{\log n}}$; c) $n^3/\log 1000n$; d) $n^{29/10}$; e) $2^n \log n/n^2$;

f) 2^{2^n}; g) $n^{2.5}\log^{10} n$; h) $n\log n\log\log n + n^3$; i) $n\log^4 n$; j) $n^{3.5}/\log\log^7 n$.

2.(15) Оценить сложность вызова $PR(a[1..N])$ для следующего алгоритма (N - степень тройки).
procedure $PR(A[m..n])$:
begin if $(n - m \leq 1)$ then return(1)
else begin $p := n - m + 1$;
обработать$(A[1..p])$; обработать$(A[m + 2p..n])$;
$PR(A[m..m + p/3 - 1]); PR(A[m + 2p/3..n])$
end end
(сложность процедуры "обработать$(x, A[m, n])$"считать равной $O((n - m)^2)$).

3. (15) Напишите нерекурсивный вариант процедуры префиксного обхода бинарного дерева. Докажите его корректность и оцените сложность.

4.(15) Как изменить алгоритм, использующий метод динамического программирования для распознавания кс-языка, порождаемого кс-грамматикой в нормальной форме Хомского, чтобы он в случае положительного ответа возвращал и дерево вывода исходного слова? Какова будет сложность нового алгоритма?

5.(15) Рассмотрим следующую программу для разбиения множества S на подмножества $S_1 = \{b|b \in S, b < a\}$ и $S_2 = \{b|b \in S, b \geq a\}$.
Вход: массив $A[1..n]$ и число a.
Выход: массив $A'[1..n]$, в котором вначале идут элементы S_1, а затем – S_2.

```
BEGIN i := 1; j := n;
    WHILE i < j + 1 DO
        BEGIN WHILE (A[j] ≥ a)AND(j > 0) DO j := j − 1;
            WHILE (A[i] < a) AND (i < n + 1) DO i := i + 1;
            IF i < j THEN BEGIN
                переставить A[i] и A[j];
                i := i + 1; j := j − 1
END END END.
```

Доказать правильность алгоритма и оценить время работы.

6.(15) Какие ответы будет давать "стратегия дьявола"из теоремы Кислицына в ответ на следующие вопросы:

$$a > b?; d > c?; d > a?; d > b?; e > b?; a > e?; f > g?; f > c?; f > e?; f > d?; e > d?$$

(если ответ не определен, пусть "побеждает"игрок, стоящий в сравнениии слева). Какие пары (X, Y) находятся в отношении "X превосходит Y"?

7.(10) Отсортировать следующую последовательность методом слияния:
12, 7,2,5,81,10,4,3,7,2,1,23,5, 44.
Определить количество вызовов процедуры СЛИ(A, B).

8.(15) Написать РАМ-программу для вычисления функции $f(n) = n^n$ с временной сложностью $O(\log n)$ при равномерном весе.

Вариант 12

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^2/\log^5 n$; b) $n^{\sqrt{n}}/n^2$; c) $n^{1.9}\log^{100} n$; d) $n^{0,1}/\log^{14} n$; e) $n^{n/\log n}$;

f) $n!/2^n$; g) $n^{0.5}\log^{10} n$; h) $n^{2\log n+\log\log n}$; i) $n^2\log^n n$; j) $n^2(\log\log n)^{10}/\log n$.

2. (10) Предложите алгоритм преобразования t-ичного дерева в эквивалентное ему бинарное дерево. Какова его сложность?

3.(15) Пусть дан массив записей, который нужно отсортировать по ключу, имеющему 2 значения 0 и 1. Предложите алгоритм такой сортировки за линейное время и с дополнительной памятью $O(1)$.

4.(15) В доказательстве теоремы Кислицына о сложности нахождения второго по величине элемента введено отношение "А превосходит В". Докажите, что оно транзитивно.

5.(20) Идея алгоритма сортировки вставками заключается в том, что на этапе j (j=2,...,n) элемент A[j] ставится на свое место среди элементов A[1], A[2], ..., A[j-1]. Докажите, что следующая программа сортирует массив A[1..n] и оцените время ее работы.

```
FOR j = 2 UNTIL n DO
  BEGIN
      x := A[j]; k := 1;
      WHILE x > A[k] AND k < j DO
              k := k + 1;
      FOR i = j UNTIL k + 1 STEP −1 DO
              A[i] := A[i − 1];
      A[k] := x
  END.
```

6. (15) Напишите алгоритм выполнения команды РАМ
STORE *5
моделирующей машиной Тьюринга. Каково максимальное время моделирования, если эта команда выполнялась РАМ на шаге t?

7.(15) Докажите, что неориентированный граф $G = (V, E)$ является деревом $\Leftrightarrow$ G - ациклический, но добавление любого ребра порождает в нем цикл.

8.(10) Используя алгоритм лексикографической сортировки, упорядочить последовательность:
РБАР, РАБА, КРАБ, АРБА,БРАК, АКАР,БАБА, АББА.
Какова теоретическая оценка сложности этого алгоритма?

Вариант 13

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^{\log n} \log^5 n$; b) $n^{\sqrt{n}}$; c) $n^3/log^{100}n$; d) $n^{299/100}$ e) $2^{n \log n}/n^2$;

f) $2^{\log^2 n}$; g) $n^{3.6}/\log^{10} n$; h) $n^3 \log n \log\log n$; i) $n^3 \log^4 n$; j) $n^{3.5} \log\log^5 n$.

2.(15) Оценить сложность вызова $PR(a[1..N])$ для следующего алгоритма (N - степень тройки).
procedure $PR(A[m..n])$:

begin if $(n - m \leq 1)$ **then return**(1)
else begin $p := n - m + 1$;
обработать$(A[1..p])$; обработать$(A[m + 2p..n])$;
$PR(A[m..m + p/3 - 1]); PR(A[m + 2p/3..n])$
end
end

(сложность процедуры "обработать$(x, A[m, n])$"считать равной $O(n - m)$).

3. (15) Напишите нерекурсивный вариант процедуры префиксного обхода бинарного дерева. Докажите его корректность и оцените сложность.

4.(15) Как изменить алгоритм, использующий метод динамического программирования для распознавания кс-языка, порождаемого кс-грамматикой в нормальной форме Хомского, чтобы он в случае положительного ответа возвращал и дерево вывода исходного слова? Какова будет сложность нового алгоритма?

5.(15) Рассмотрим следующую программу для разбиения множества S на подмножества $S_1 = \{b|b \in S, b < a\}$ и $S_2 = \{b|b \in S, b \geq a\}$.
Вход: массив $A[1..n]$ и число a.
Выход: массив $A'[1..n]$, в котором вначале идут элементы S_1, а затем – S_2.

```
BEGIN i := 1; j := n;
    WHILE i < j + 1 DO
        BEGIN WHILE (A[j] ≥ a)AND(j > 0) DO j := j − 1;
            WHILE (A[i] < a) AND (i < n + 1) DO i := i + 1;
            IF i < j THEN BEGIN
                переставить A[i] и A[j];
                i := i + 1; j := j − 1
END END END.
```

Доказать правильность алгоритма и оценить время работы.

6. (15) Пусть исходный массив А уже отсортирован в порядке возрастания. Каким будет время его сортировки с помощью дерева? А каким оно будет, если он отсортирован в порядке убывания?

7 . (10) Начертить двоичное дерево, соответствующее формуле
$((a + b) * (w - s) * (c + x)) + (d - (f + g * (n + v)))$. Выписать его узлы в префиксном и в инфиксном порядках.

8. (10) Используя алгоритм лексикографической сортировки, упорядочить последовательность: ВОЛ, ВОВА, АВОЛ, ЛАО, ЛОЛА, ВОПВ, ЛАВ,ПОЛ.

Вариант 14

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^5 \log^6 n$; b) $2^{\sqrt{n}}/n^5$; c) $n^6/\log^{100} n$; d) $2^{n \log n}$; e) $n^{51/10}/\log^9 n$;

f) $(n!)^{\log n!}$; g) $n^{0.5} \log^{10} n$; h) $n \log n \log \log n$; i) 2^{2^n}; j) $n^{5.01} \log^5 n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$: следующей процедуры (n - степень 5.)
procedure PEL(A[ш..n]):

begin
$p := (n - m + 1)/5; q := (n - m + 1)/2;$
$MAKE(A[m..n]);$
$PEL(A[m..m + p - 1]); PEL(A[m + p..m + 2p - 1]); PEL(A[m + 2p..n]);$
$PEL(A[m + q..m + q + p - 1]); PEL(A[m + q - p..m + q - 1])$
end

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $O((n - m)^2)$.

3.(15) Докажите, что неориентированный граф $G = (V, E)$ является деревом $\Leftrightarrow$ G — ациклический, но добавление любого ребра порождает цикл.

4.(20) Назовем элементарным преобразованием слова удаление буквы, вставку буквы или изменение одной буквы слова. Написать алгоритм, который по любой паре слов u, v подсчитывает минимальное число элементарных преобразований, необходимых для преобразования слова u в слово v и находит минимальную цепочку преобразований. Оценить его сложность.

5. (15) Модифицировать алгоритм быстрой сортировки так, чтобы он работал за время $O(n \log n)$. (Указание: используйте идею алгоритма МИНК).

6.(15) В док-ве теоремы Кислицына о сложности нахождения второго по величине элемента введено отношение "A превосходит B". Докажите, что "чемпион"превосходит всех остальных игроков. (Используйте индукцию по числу игроков).

7 (10) . Упорядочить последовательность $31, 16, 23, 14, 11, 13, 8, 32, 11$, применяя алгоритм сортировки деревом. Сколько перемещений элементов производится в процессе сортировки?

8 (10). Начертить двоичное дерево, соответствующее формуле
$(((a + b) + (w - s) * c) + x) - (d - n * (f + g * (x + v)))$.
Выписать его узлы в префиксном и в суффиксном порядках.

Вариант 15

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^3/\log^4 n$; b) $n^{\sqrt{2n}}/n^2$; c) $n^{2.99}\log^{10} n$; d) $n^{3.1}\log^4 n$; e) $n^{n\log n}$;

f) $n!/3^n$; g) $n^{3.05}\log^{10} n$; h) $n^3\log n\log\log^2 n$; i) $2^n\log^n n$; j) $n^3(\log\log n)^{10}/logn$.

2.(15) Оценить сложность вызова PEL(A[1..n]): следующей процедуры. (n - степень 4) procedure $PEL(A[m..n])$:

begin $p := (n-m+1)/4; q := (n-m+1)/2;$
$MAKE(A[m..n]);$
$PEL(A[m..m+p-1]); PEL(A[m+p..m+2p-1]); PEL(A[m+2p..n]);$
$PEL(A[m+q..m+q+p-1]); PEL(A[m+q-p..m+q-1])$
end

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $O(n-m)$.

3. (15) Пусть $G = (V, E)$ - ориентированный ациклический граф, в котором из некоторой вершины v в любую другую вершину u из V ведет единственный путь. Покажите, что неориентированный вариант графа G является деревом.

4.(20) Даны две строки x и y. Строка x состоит из нулей и единиц, строка y из символов A и B. Можно ли строку x преобразовать в строку y по следующему правилу: цифра 0 преобразуется в непустую последовательность букв A, а цифра 1 - либо в непустую последовательность букв A, либо в непустую последовательность букв B? Постройте алгоритм, отвечающий на этот вопрос и оцените его сложность.
Указание. Пусть $C[i,j] = 1 \Leftrightarrow$ i символов последовательности x можно преобразовать в j символов последовательности y. Предложите алгоритм для вычисления массива C размера $|x| \times |y|$.

5. (15) Разработайте алгоритм, который по заданному числу k находит в данном множестве S мощности n его k элементов, ближе всего расположенных к медиане. Сложность алгоритма должна быть $O(n)$.

6.(15) Модифицировать алгоритм МИНК так, чтобы первоначальная последовательность разбивалась на отрезки длины 7 (а не 5). Оценить сложность получившегося алгоритма. Можно ли получить линейный алгоритм при разбиении на отрезки длины 3?

7. (10). Построить для дерева T, заданного списком дуг, представляющее его бинарное дерево T_1. Построить инфиксный обход T_1.
$T = \{(v1, v2), (v1, v3), (v1, v7), (v1, v4), (v2, v5), (v2, v6), (v3, v8), (v3, v9), (v3, v10), (v5, v11)\}$.

8 (10). Используя алгоритм лексикографической сортировки, упорядочить последовательность: ВАВА, ВОВА, АВОЛ, АЛАО, ЛОЛА, ОПОП, ЛАВА, ПОЛО.

Вариант 16

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
) $2^{n/\log^2 n}$; b) $n^{\sqrt{n+\log n}}$; c) $n^4/\log 1000n$; d) $n^{39/10}$; e) $2^n \log n/n^2$;

f) $2^{\log n!}$; g) $n^{1.5}\log^{10} n$; h) $n^2 \log n \log\log n + n^3$; i) $n^3 \log^4 n$; j) $n^{3.5}/\log\log^7 n$.

2.(15) Оценить сложность вызова $PEL(A[1..n])$: следующей процедуры (n - степень тройки).
procedure$PEL(A[m..n])$:
{
$p := [(n-m+1)/3]; q := [(n-m+1)/2];$
$MAKE(A[m..n]);$
$PEL(A[m..m+p-1]); PEL(A[m+p..m+2p-1]);$
}

Считать сложность вызова процедуры $MAKE(A[m..n])$ равной $\theta((n-m)\log(n-m))$.

3. (15) Пусть $f(n)$ и $g(n)$ – это две неубывающие функции, ограниченные полиномами. Предположим, что для некоторой геометрической прогрессии $n_1 < n_2 < ...n_i < ...$ имеет место соотношение $O(f(n_i)) = O(g(n_i))$. Доказать, что тогда $O(f(n)) = O(g(n))$. Показать, что это утверждение неверно для немонотонных функций.

4.(20) Назовем общей подпоследовательностью двух слов $V = v(1)v(2)...v(n)$ и $W = w(1)w(2)...w(m)$ последовательность индексов $1 \le i_1 < \ldots < i_k \le n$ и последовательность индексов $1 \le j_1 < \ldots < j_k \le m$ таких, что $v(i_1) = w(j_1), \ldots, v(i_k) = w(j_k)$. Используя динамическое программирование, предложите алгоритм нахождения самой длинной общей подпоследовательности и оцените его сложность.

5.(15) Докажите, что сортирующее дерево из n элементов содержит не более $[n/2(h+1)]$ вершин высоты h.

6.(15) Какие ответы будет давать "стратегия дьявола"из теоремы Кислицына в ответ на следующие вопросы:
$a > b?; d > c?; d > a?; d > b?; e > b?; a > e?; f > g?; f > c?; f > e?; f > d?; e > d?$
(если ответ не определен, пусть "побеждает"игрок, стоящий в сравнениии слева). Какие пары (X, Y) находятся в отношении "X превосходит Y"?

7. (10) Построить линейную программу, вычисляющую определитель 3 x 3 матрицы по ее элементам. Оценить время ее работы и используемую память.

8. (10) Упорядочить последовательность 23,16,23,14,11,3,8,32,11, 8, применяя алгоритм быстрой сортировки (разбиение массива производить относительно среднего арифметического его концов). Сколько сравнений производится ?

Вариант 17

1. (20) Расположите следующие функции в порядке возрастания (при $n \to \infty$):
a) $n^3 \log n \log\log n + n^4$; b) $n^{\sqrt{\log n}}$; c) $n^4/\log 1000n$; d) $n^{41/10}$; e) $2^n \log n/n^2$;

f) 3^{2^n}; g) $n^{3.75} \log^{10} n$; h) $3^{n/\log n}$; i) $(\log n!)^4/\log\log^5 n$; j) $2^{n^{3.5}}/\log^7 n$.

2. (20) Пусть $f(n)$ и $g(n)$ - неубывающие функции и для всех n выполнены следующие неравенства:
a) $4g(n) < g(2n) < 8g(n)$;
b) $f(2n) < 2f(n) + g(n)$.
Доказать, что $f(n) \in O(g(n))$.

3. (10) Доказать, что время работы линейного алгоритма МИНК для выбора k-го наименьшего элемента $T(n) \leq 20cn$, где cn - время его работы при $n \leq 50$.

4. (20) Пусть функция "веса"$t(x, y)$ задана для пар символов алфавита $A \cup \{-\}$ следующими равенствами:$t(x, x) = 1, t(-, x) = t(x, -) = -1, t(x, y) = -2(x \neq y)$, $x, y \in A$. Модифицируйте алгоритм оценки оптимального выравнивания двух слов в алфавите A так, чтобы он находил не только вес оптимального выравнивания, но и само это выравнивание.

5. (15) Пусть T - двоичное дерево поиска, с различными метками всех вершин. Пусть x - его лист, а y - отец вершины x в T. Покажите, что метки $l(x)$ и $l(y)$ являются соседями в упорядоченном списке всех меток T.

6.(15) Доказать правильность алгоритма КОД, строящего оптимальное дерево двоичного поиска (т.е. показать индукцией по l правильность определения стоимостей $c_{i,j}$ и корней $r_{i,j}$).

7.(10) Найти медиану следующей последовательности с помощью алгоритма линейной сложности:

17, 12 ,5, 81,15, 4, 3, 7, 2, 1,17, 8, 9, 5, 26, 23, 11, 13, -2, 0, 41, 15, 20, 17, 5, 12, 3.

8.(10) Отсортировать следующую последовательность методом слияния.
17, 2, 15, 8, 10, 4, 13, 7, 2, 1, 23, 5.
Определить количество вызовов процедуры СЛИ(A, B).