

## Контрольная № 1 (дискретная математика для программистов) (февраль-март-2024)

### Вариант 71

1. Сколькими способами из 28 костей домино можно выбрать две кости так, чтобы их можно было приложить друг к другу?

2. Найдите решение рекуррентного соотношения:

$$a_{n+2} - 5a_{n+1} + 6a_n = 0, a_1 = 1, a_2 = -7 \text{ и постройте порождающую функцию.}$$

3. Символы 7-ми буквенного алфавита имеют частоты:

(A 1), (B 8), (C 9), (D 16), (E 17), (F 22), (G 27)

Сколько в среднем бит информации несет один символ такого сообщения?

4. Для условий предыдущего задания сконструировать коды Шеннона-Фано и для них вычислить среднюю информационную емкость одного символа сообщения.

5. Две кодовые строки построены по методу Хемминга. Известно, что в каждой из них не более одной ошибки. Восстановить исходные кодовые строки.

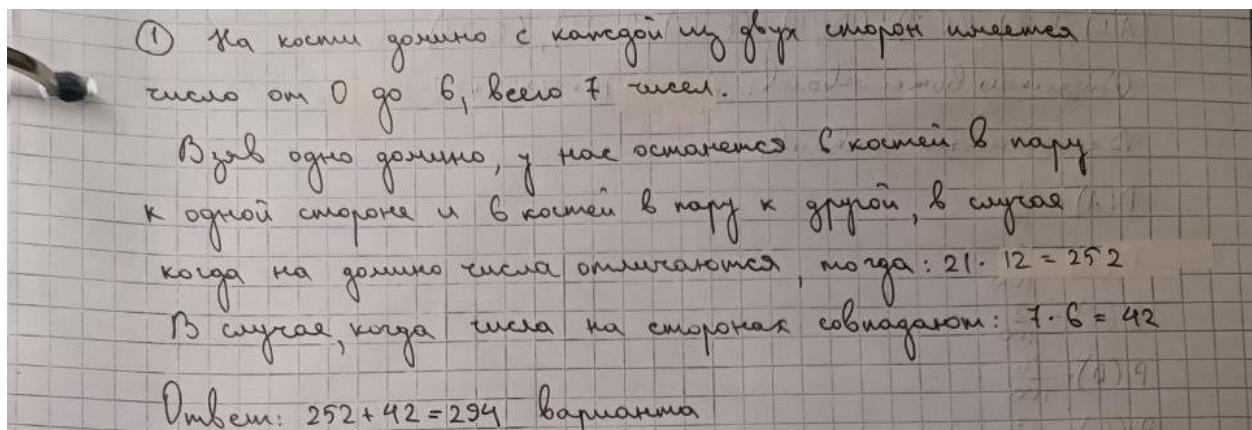
1 0 0 0 0 0 1 0 0 1 0 0 0 0 1

0 0 0 0 0 1 1 0 0 1 0 1 1 1 1

6. Дать определение (или описание) и привести три примера. Тривиальные, полные графы.

7. Дать определение (или описание) и привести три примера. Сжатие информации алгоритмом Хаффмана.

### Задание 1



## Задание 2

$$(2) \quad a_{n+2} - 5a_{n+1} + 6a_n = 0, \quad a_1 = 1, \quad a_2 = -7$$

$$1) \quad p(\lambda) = \lambda^2 - 5\lambda + 6 = 0$$

$$\lambda_1 = 2, \quad \lambda_2 = 3$$

$$2) \quad a_n = C_1(2)^n + C_2(3)^n$$

$$\begin{cases} C_1 + C_2 = 1 \\ C_1(2)^1 + C_2(3)^1 = -7 \end{cases}$$

$$C_1 = -4, \quad C_2 = 5$$

$$3) \quad a_n = -4(2)^n + 5(3)^n, \quad a_0 = -4 + 5 = 1$$

$$\text{Ответ: } F(t) = \frac{a_0 + (a_1 + pa_0)t}{1 + pt + qt^2} = \frac{1 + (1 + (-5) \cdot 1)t}{1 + (-5)t + 6t^2} = \frac{1 - 4t}{1 - 5t + 6t^2}$$

### Задание 3

⑤ Дан 7-ми буквенный алфавит:

(A 1), (B 8), (C 9), (D 16), (E 17), (F 22), (G 27)

Общая частота символов:  $1+8+9+16+17+22+27=100$

Вероятности символов:

$$P(A) = \frac{1}{100}$$
$$P(B) = \frac{8}{100}$$
$$P(C) = \frac{9}{100}$$
$$P(D) = \frac{16}{100}$$
$$P(E) = \frac{17}{100}$$
$$P(F) = \frac{22}{100}$$
$$P(G) = \frac{27}{100}$$

По формуле Шеннона:  $I_x = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_n \log_2 p_n)$

Подставив, получим:

$$I = (1/100) * 6.6439 + (8/100) * 3.3219 + (9/100) * 3.1699 + (16/100) * 2.6439 + (17/100) * 2.5849 + (22/100) * 2.4426 + (27/100) * 2.3700 \approx 2.6090 \text{ бит}$$

## Задание 5

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1
3	0	0	0	0	0	1	1	0	0	1	0	1	1	1	1

Q	R	S	T
Позиция 1:	Позиция 2:	Позиция 4:	Позиция 8:
3	3	2	2
3	5	6	5

1) Ошибка в позиции  $1+2=3$

2) Ошибка в позиции  $1+2+8=11$

[illegible]

## Задание 6

Тривиальный граф — это структура данных графа, в которой присутствует только одна вершина или узел без рёбер.

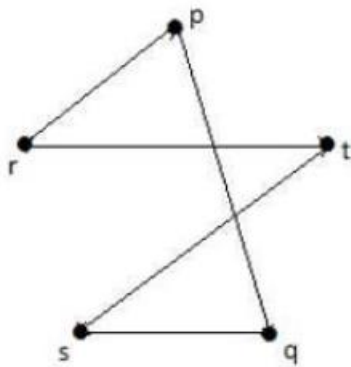
Полный граф - это простой неориентированный граф, в котором каждая пара вершин соединена ребром. Другими словами, в полном графе каждая вершина связана с каждой другой вершиной.

Примеры:

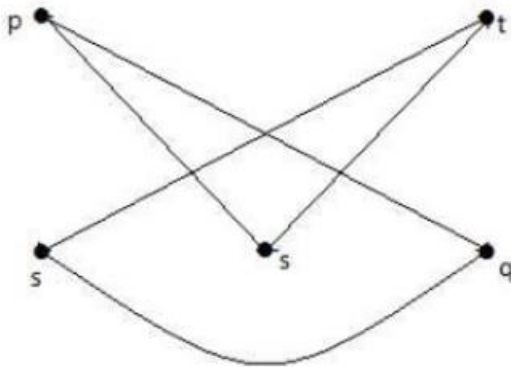
1)



2)



3)



## Задание 7

Описание алгоритма Хаффмана:

Сжатие информации алгоритмом Хаффмана - это алгоритм сжатия данных, используется для сокращения размера файла или передаваемых данных путем присвоения более коротких кодов часто встречающимся символам и более длинных кодов реже встречающимся символам. Алгоритм Хаффмана основан на построении оптимального префиксного кода, где код каждого символа не является префиксом кодов других символов.

Шаги алгоритма Хаффмана обычно включают:

- 1) Анализ частоты символов: Определение частоты встречаемости каждого символа в исходных данных или файле, который необходимо сжать.
- 2) Построение дерева Хаффмана: Создание дерева Хаффмана, где каждый символ представлен в виде листа дерева, а более часто встречающиеся символы находятся ближе к корню дерева.
- 3) Присвоение кодов: Присвоение двоичных кодов каждому символу, где путь от корня к каждому символу представляет код символа. Часто встречаемым символам присваиваются более короткие коды, а реже встречаемым символам присваиваются более длинные коды.
- 4) Создание сжатого файла: Замена каждого символа в исходных данных его соответствующим кодом Хаффмана. Это позволяет сжать данные, заменяя более длинные последовательности символов более короткими кодами.

Примеры:

- 1) Сжатие текстового файла: Замена часто встречающихся символов более короткими кодами и реже встречающихся символов более длинными кодами.
- 2) Сжатие изображения: Кодирование символов, представляющих пиксели изображения. Часто встречающиеся цвета или пиксели могут быть закодированы более короткими кодами, что позволяет сжать изображение без значительной потери качества.

3) Сжатие аудиофайла: Часто встречающиеся звуковые сэмплы могут быть закодированы более короткими кодами, позволяя уменьшить размер файла без значительной потери качества звука.