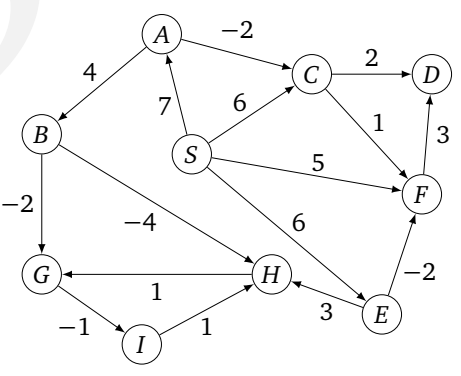
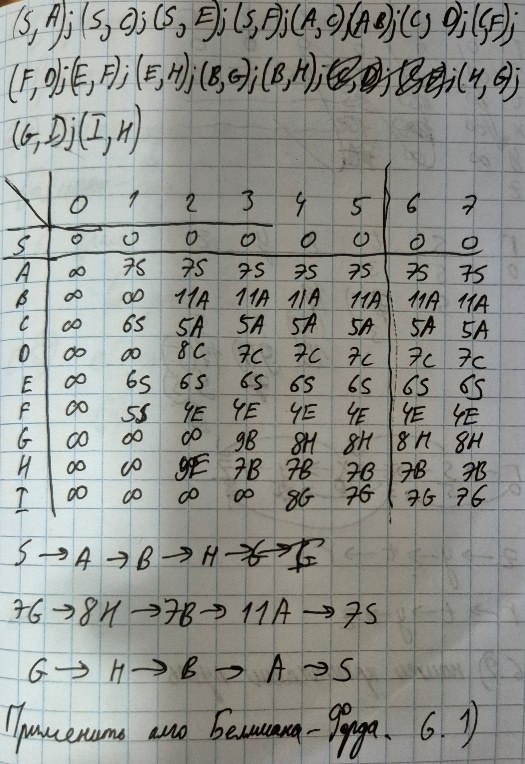
6.1) Применить алгоритм Беллмана-Форда для графа с отрицательными весами. S –стартовая вершина



Нарисовать таблицу и изобразить дерево кратчайших путей. 

6.2) Выполнить алгоритм поиска кратчайших путей в ациклическом графе:

1

6

r

s

t

x

y

z

2

5

-2

-1

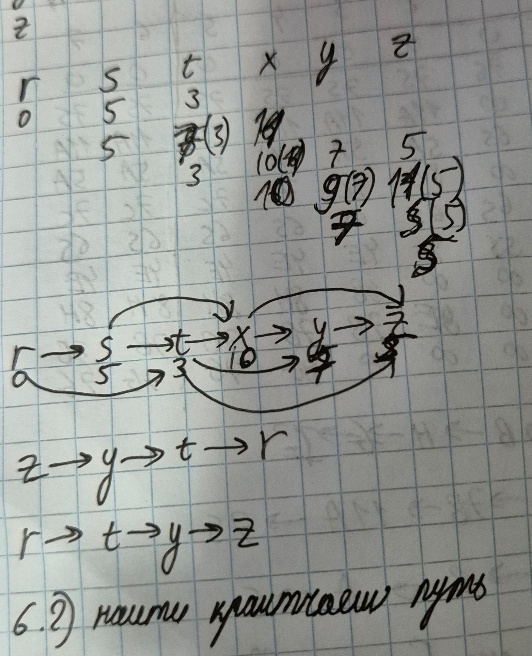
7

7

2

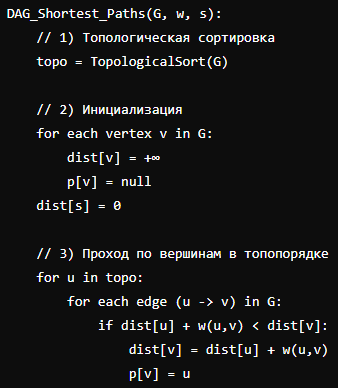
3

43

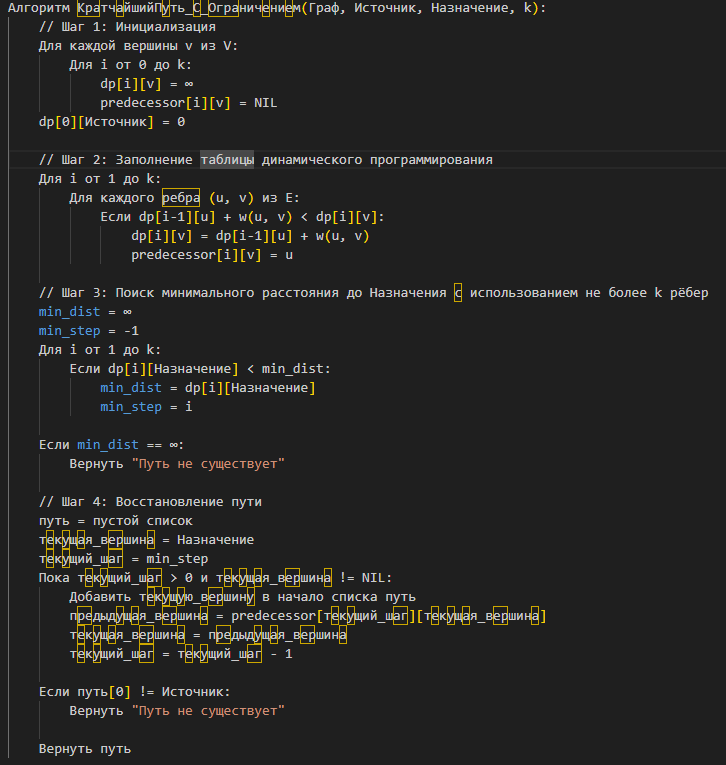


6.3) Записать алгоритм поиска кратчайшего пути в ациклическом графе (на основе топологической сортировки). Оценить время работы

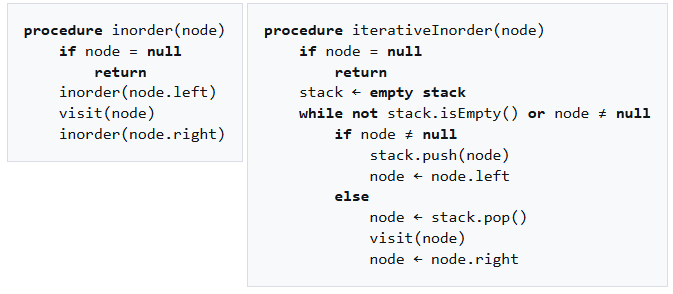
Время: O(V+E)



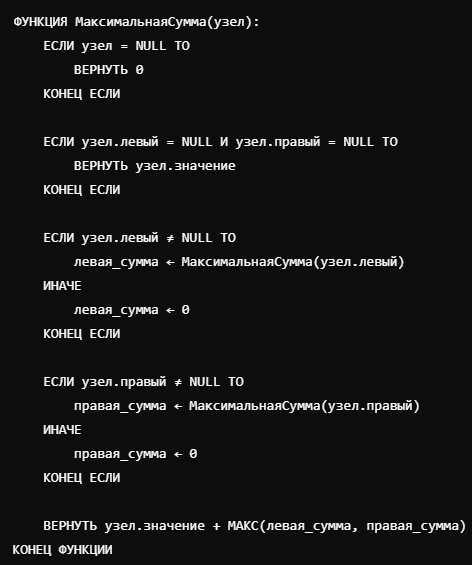
6.4) Дан ориентированный граф (возможно, с отрицательными весами на ребрах). Известно, что для двух его вершин найдется кратчайший путь из первой во вторую не более, чем *k* ребер. Постройте алгоритм, находящий этот путь за время .



6.5) Написать процедуру симметричного обхода бинарного дерева поиска. Оценить время работы. **Время: O(n)**



6.6) Найти путь с максимальной суммой, который идет от корня и до листа. (Все узлы имеют положительные ключи). Составить алгоритм и оценить сложность



7.1) Важно, чтобы хеш-функция обеспечивала хорошее распределение (как можно шире – в идеале – равномерно).

Пусть имеются 4 хеш-функции

А) возвращает «1» для .

B) возвращает длину строки в качестве индекса.

С) возвращает первый символ строки в качестве индекса.

D) каждой букве ставится в соответствие простое число(A=2, Б=3, В=5, Г=7, Д=11 и т.д. Для строки – остаток от деления суммы всех значений на размер хеша .)

Например, если , =ГДЕ → .

В каком из примеров хеш-функции будут обеспечивать хорошее распределение?

* 1. Телефонная книга, в которой ключами являются имена, а значениями – номера телефонов. Список: Анна, Владимир, Вениамин, Алла**. Ответ:**
  2. Связь батарейки с напряжением: А, АА, ААА, АААА**: Отве: ,**
  3. Связь названий книг с именами авторов. Наименования книг: «Анна Каренина», «Авиатор», «Борис Годунов». **Ответ:**

7.2) Заполнить таблицу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Время  Операции | Хеш-таблица  (средний случай) | Хеш-таблица  (худший случай) | Массивы | Связный список |
| Поиск  (Просмотреть) | O(1) | O(n) | O(1) | O(n) |
| Вставить | O(1) | O(n) | O(n) | O(1) |
| Удалить | O(1) | O(n) | O(n) | O(1) |

7.3) Продемонстрируйте происходящее при вставке в Х-Т с разрешением коллизий методом цепочек ключей 5, 28, 19, 15, 20, 33, 12, 17, 19.

|  |  |
| --- | --- |
| 0 | - |
| 1 | 17 -> 33 |
| 2 | - |
| 3 | 19->19 |
| 4 | 12->20->28 |
| 5 | 5 |
| 6 | - |
| 7 | 15 |

7.4) Пусть коэффициент заполнения Х-Т Если все элементы хешируются по ячейкам равномерно, то средняя длина цепочки равна Можно доказать, что среднее время удачного и неудачного поиска равно . Как изменится среднее время удачного поиска, неудачного поиска, вставки и удаления, если поддерживать списки в упорядоченном состоянии?

**Поиск (удачный)**: Θ(α)

**Поиск (неудачный)**: Θ(α)

**Вставка**: Θ(α) (вместо O(1) для неупорядоченного списка)

**Удаление** (включая поиск): Θ(α)

7.5) Рассмотрим вставку ключей 10, 22, 31, 4, 15, 28, 17, 88, 59 в Х-Т длиной с открытой адресацией. Продемонстрируйте реализацию вставки приведенного списка ключей при использовании а) линейного зондирования б) двойного хеширования

|  |  |
| --- | --- |
| 0 | 22 |
| 1 | 88 |
| 2 | - |
| 3 | - |
| 4 | 4 |
| 5 | 15 |
| 6 | 28 |
| 7 | 17 |
| 8 | 59 |
| 9 | 31 |
| 10 | 10 |
|  |  |

b)

|  |  |
| --- | --- |
| 0 | 22 |
| 1 | - |
| 2 | 59 |
| 3 | 17 |
| 4 | 4 |
| 5 | 15 |
| 6 | 28 |
| 7 | 88 |
| 8 | - |
| 9 | 31 |
| 10 | 10 |

8.1) Задача на использование хеш-таблицы с прямой адресацией.

«Счастливый номер». Напишите алгоритм, который определяет, является ли число «счастливым числом». «Счастливое число» определяется так: для целого положительного числа каждый раз число заменяется суммой квадратов чисел в каждой позиции (в каждом разряде), затем повторяют процесс, пока число не станет равным 1. Если число не является счастливым, то появится бесконечный цикл.

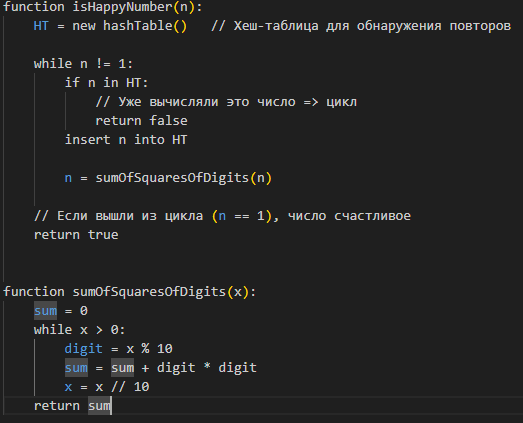
Пример: n=19 – счастливое (на выходе true)

– счастливое

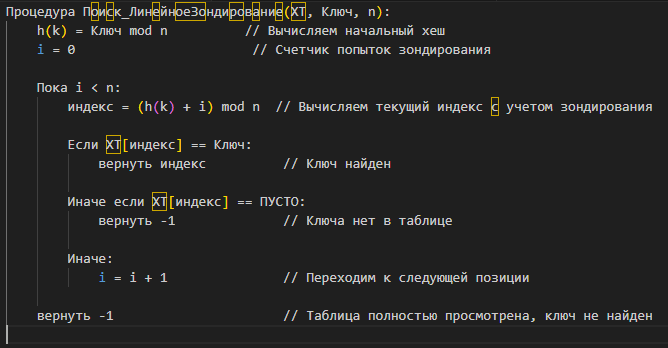
Пример: n=18 (на выходе false)

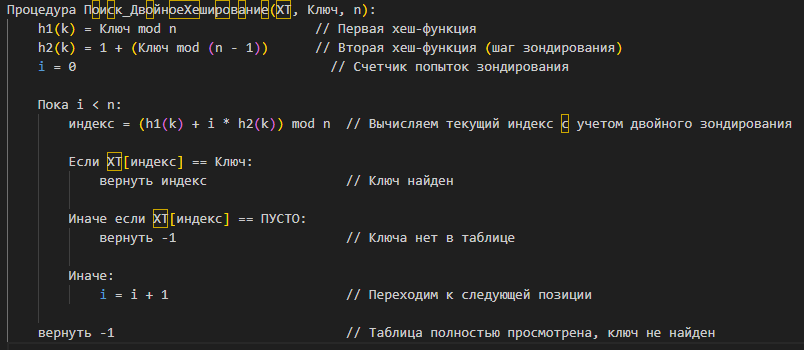
цикл

Следует сохранять в Х-Т результат каждый раз, когда вычисляется сумма квадратов.

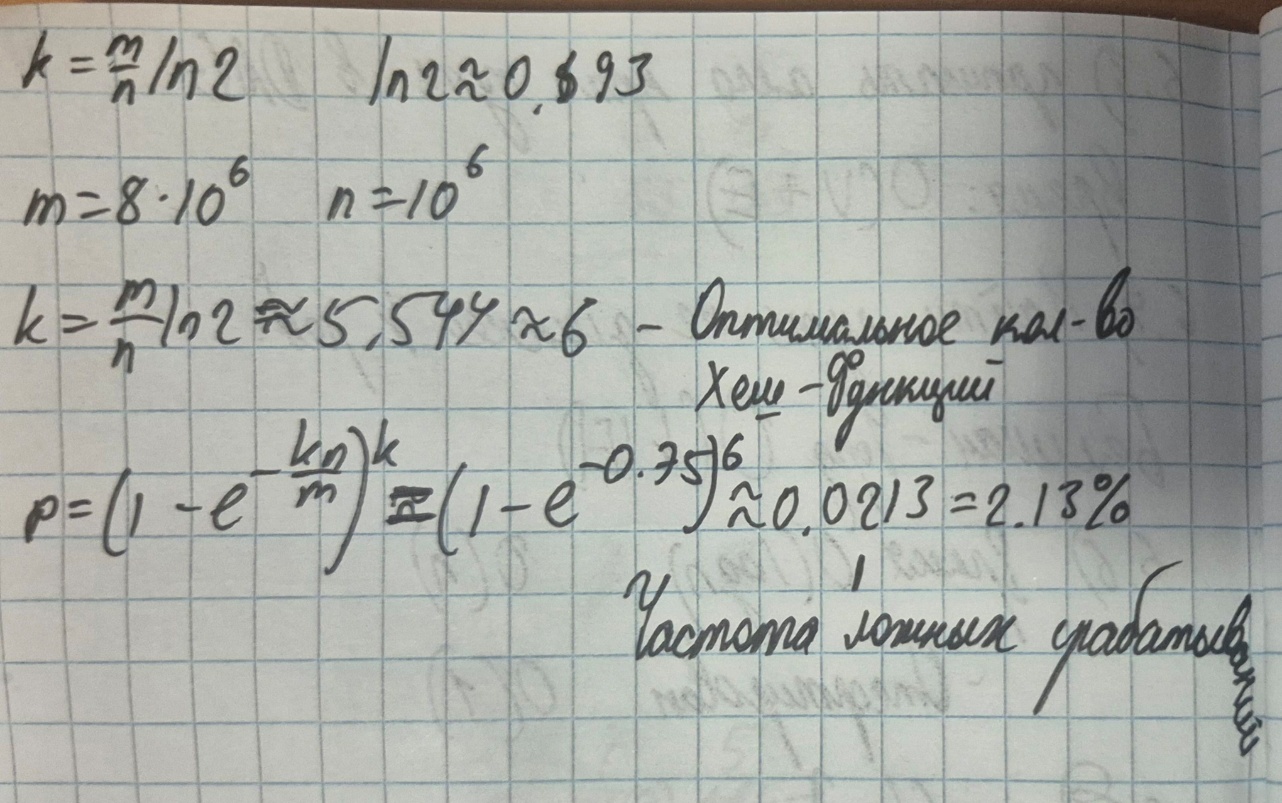


8.2) Составить псевдокод процедуры поиска в Х-Т, использующий открытую адресацию**(не понял что делать. И вот 😊)**





8.3) Вычислить оптимальное количество хеш-функций и частоту ложных срабатываний при размере фильтра Блума Мб (бит) и количестве вставленных элементов



9.1) Вы работаете в фирме по производству мебели и поставляете мебель по всей стране. Коробки с мебелью размещаются в грузовике. Все коробки имеют разный размер, и вы стараетесь наиболее эффективно использовать доступное пространство. Как выбрать коробки для того, чтобы загрузка имела максимальную эффективность? Предложите жадную стратегию. Будет ли полученное решение оптимальным?

Жадная стратегия: **Описание Стратегии:** **Сортировка коробок:** Начните с сортировки всех доступных коробок по убыванию их размеров (от самых больших к самым маленьким). Размер коробки может определяться по объёму, площади основания или другой подходящей метрике, в зависимости от специфики загрузки.

**Последовательная загрузка:** **Итерация по коробкам:** Начните с самой большой коробки и попытайтесь разместить её в грузовике.

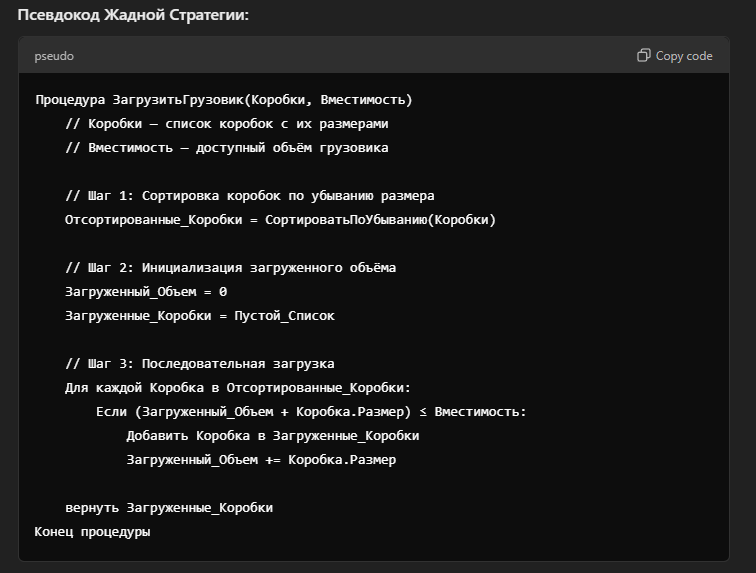
**Проверка доступного пространства:** Для каждой коробки проверьте, есть ли достаточно места в текущем грузовике для её размещения.

**Размещение или отбрасывание:**

Если место есть, разместите коробку и уменьшите доступное пространство соответствующим образом.

Если места недостаточно, перейдите к следующей коробке в списке.

**Повторение:**Продолжайте процесс до тех пор, пока все коробки не будут рассмотрены или пока грузовик не будет полностью загружен.



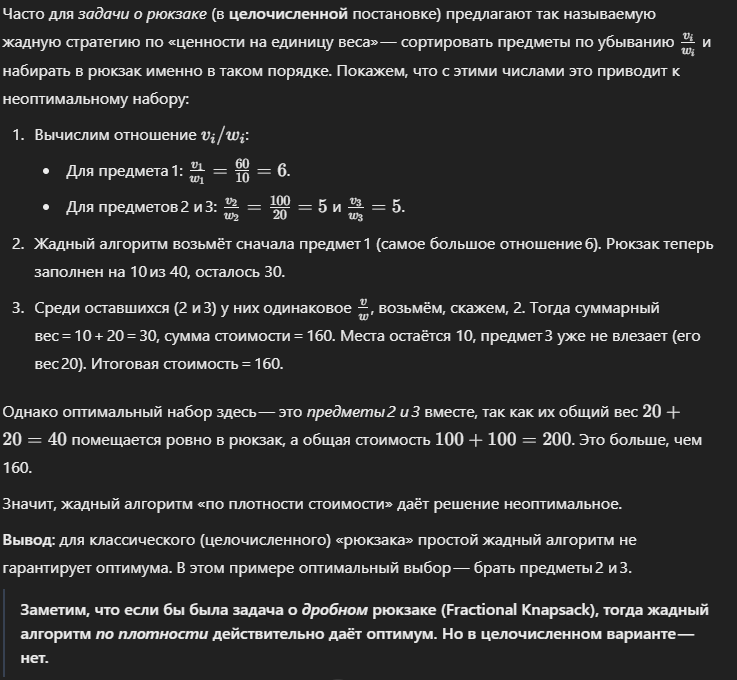
**Динамическое программирование:** Обеспечивает оптимальные решения, но требует больше вычислительных ресурсов.

**Поиск полного перебора:** Тоже гарантирует оптимальность, но непрактичен для большого количества коробок.

**Методы ветвей и границ:** Компромисс между эффективностью и оптимальностью.

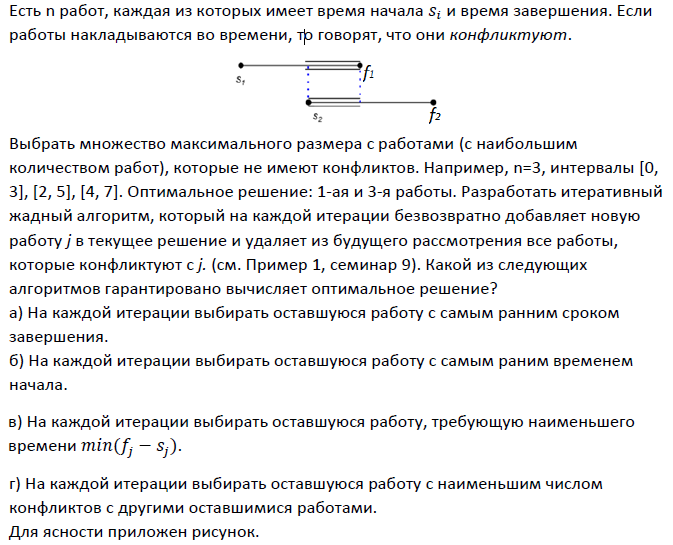
9.2) Задача о рюкзаке. Имеется N предметов, известны 𝑣𝑖 – стоимость 𝑖-го предмета, 𝑤𝑖 – масса (вес) 𝑖-го предмета. Найти набор предметов с наибольшей суммарной стоимостью и суммарной массой, не превосходящей 𝑊. Предложить жадный алгоритм и применить его для следующих данных:

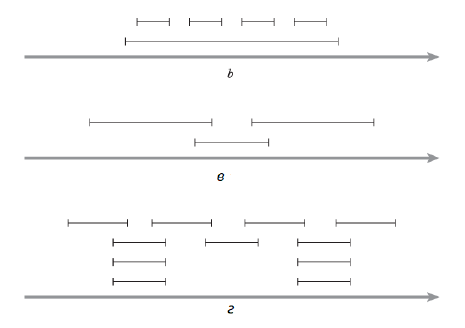
N=3, 𝑊=40, 𝑤1=10, 𝑤2=20,𝑤3=20 𝑣1=60,𝑣2=100,𝑣3=100



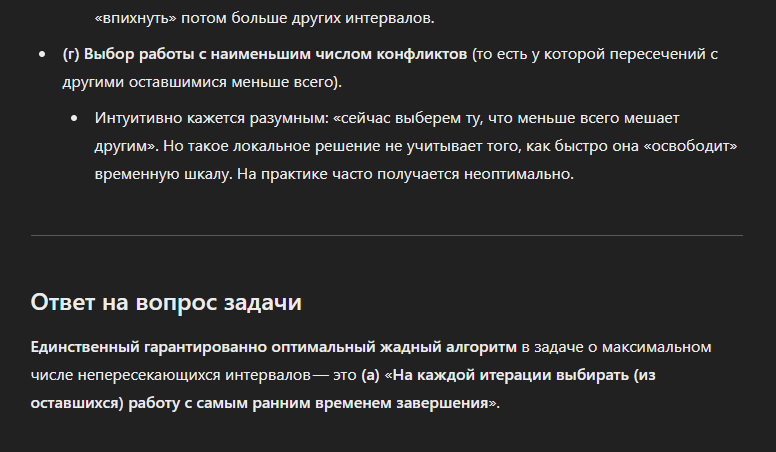
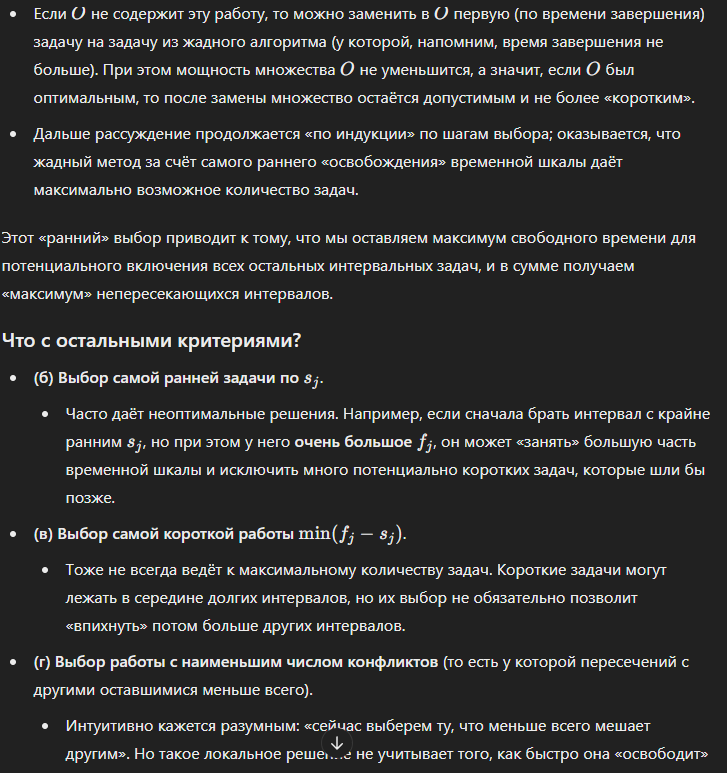
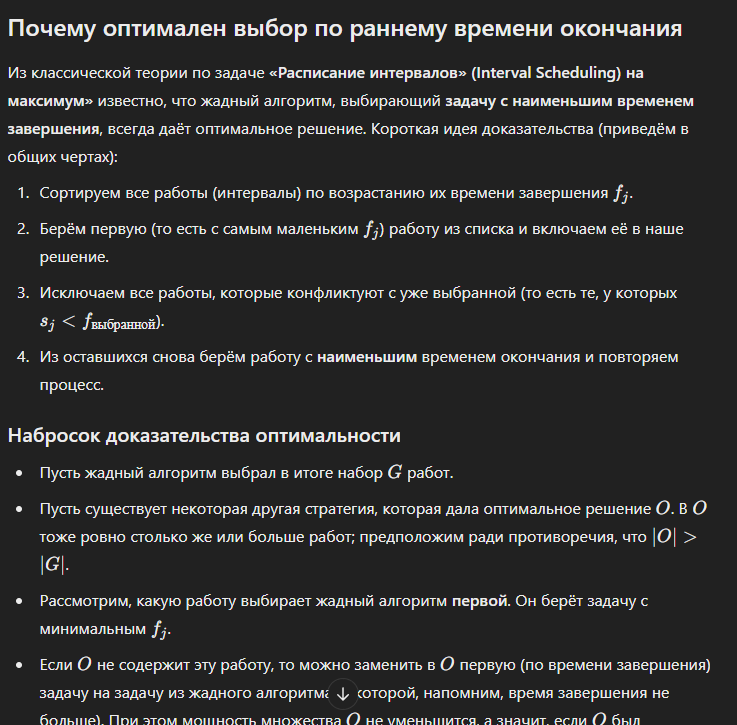
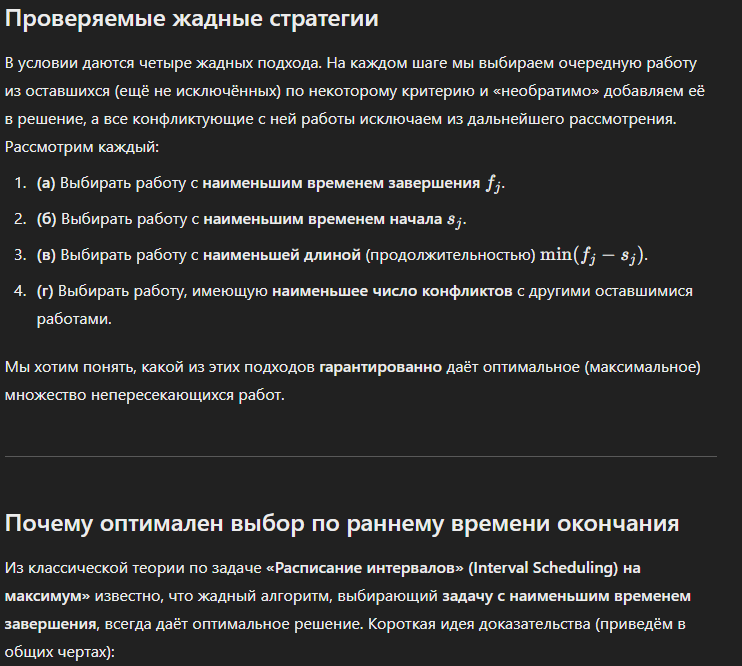
**(Don’t know, Don’t understand what to do here too)**

9.3)

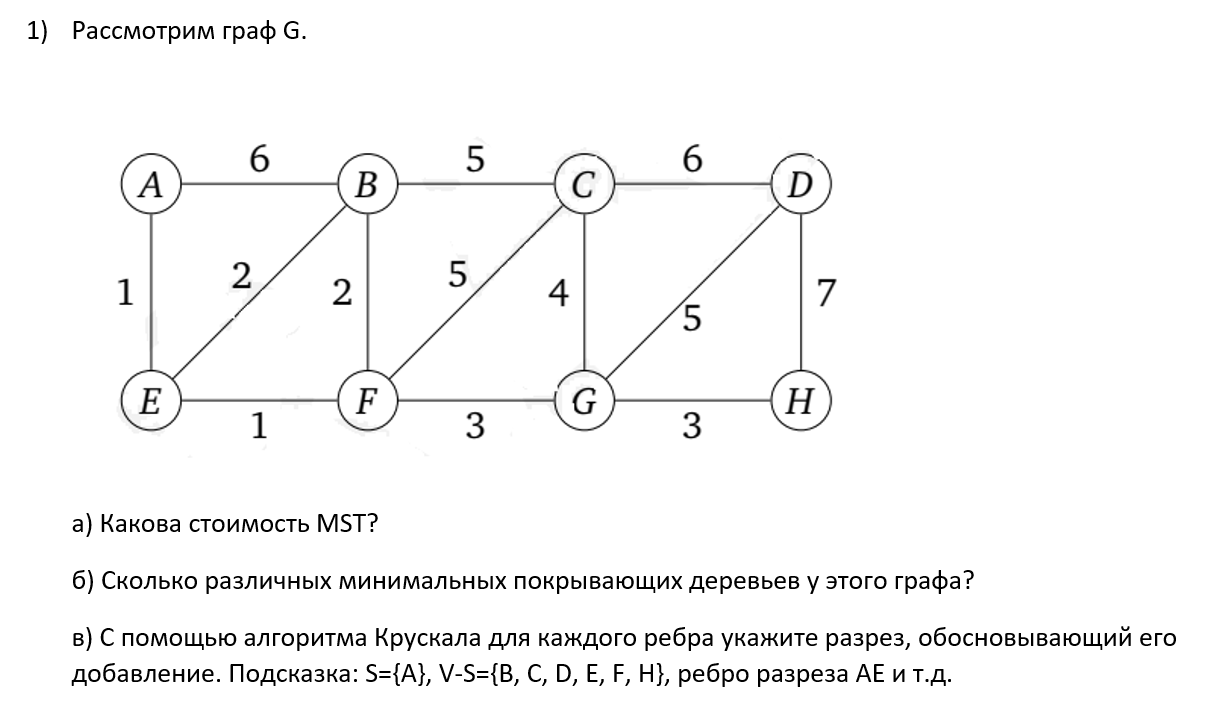


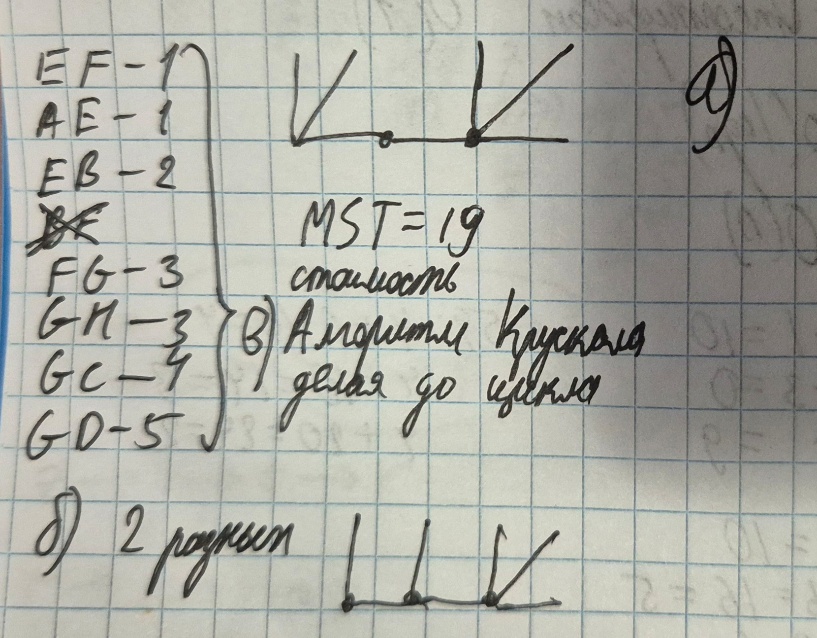


**(Ничего не понял. Ну вот ГПТ)**

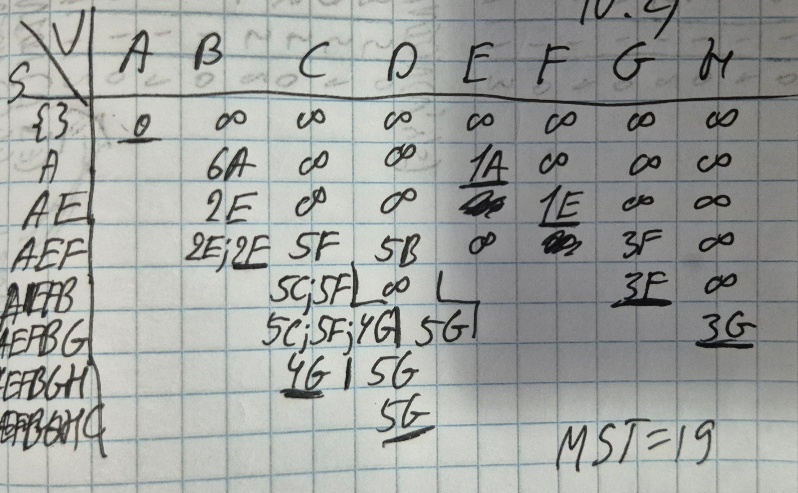


10.1)





10.2) Рассмотрим неориентированный граф из №1. Применить к нему алгоритм Прима. Когда можно выбрать несколько вершин, используйте первую по алфавиту. В частности, запустите алгоритм из вершины A. Заполните таблицу, показывающую промежуточные значения массива стоимостей (см. таблицу из последнего примера семинара 10).

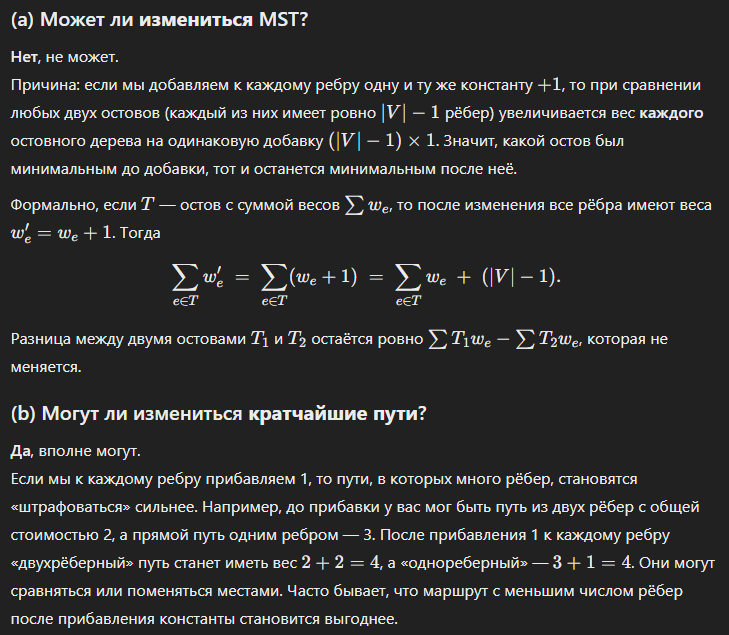


10.3) Рассмотрим неориентированный граф G=(V, E) с неотрицательными весами на ребрах .

Допустим, что мы уже нашли MST для G, а также кратчайшие пути до вершины . Пусть теперь веса всех ребер увеличились на 1 (т.е. новый вес ребра e ).

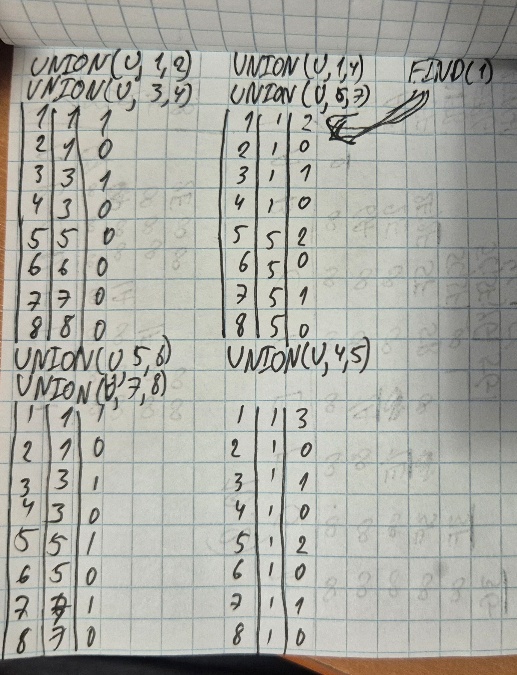
а) Может ли измениться MST (Докажите, что не может или приведите контрпример).

б) Могут ли измениться кратчайшие пути? (Докажите, что не могут или приведите пример, когда меняются).



10.4) Проследите работу реализации структуры UNION-FIND над множествами с ключами {1,2,3,4,5,6,7,8} =U для последовательных операций

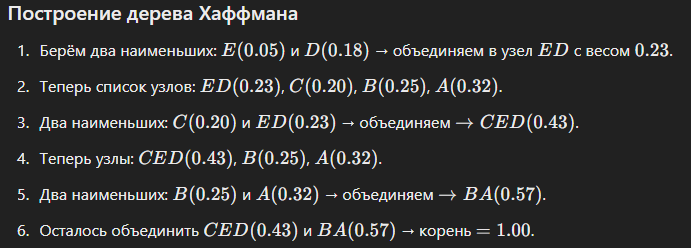
UNION (U, 1, 2); UNION (U, 3, 4); UNION (U, 5, 6); UNION (U, 7, 8); UNION (U, 1, 4); UNION (U, 6, 7); UNION (U, 4, 5); FIND (1)

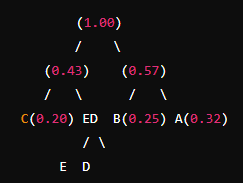


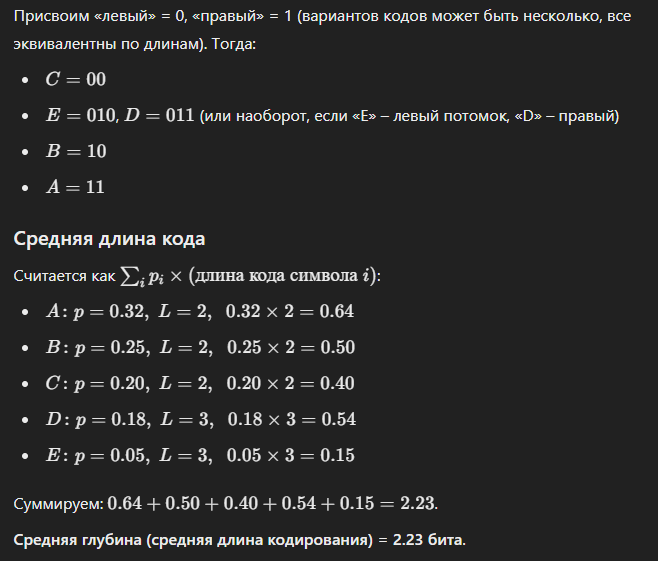
10.5) Рассмотрим следующие частоты символов для пятисимвольного алфавита:

|  |  |
| --- | --- |
| *Символ* | *Частота* |
| *A* | *0,32* |
| *B* | *0,25* |
| *C* | *0,2* |
| *D* | *0,18* |
| *E* | *0,05* |

Какова средняя глубина кодирования оптимального префиксного кода?





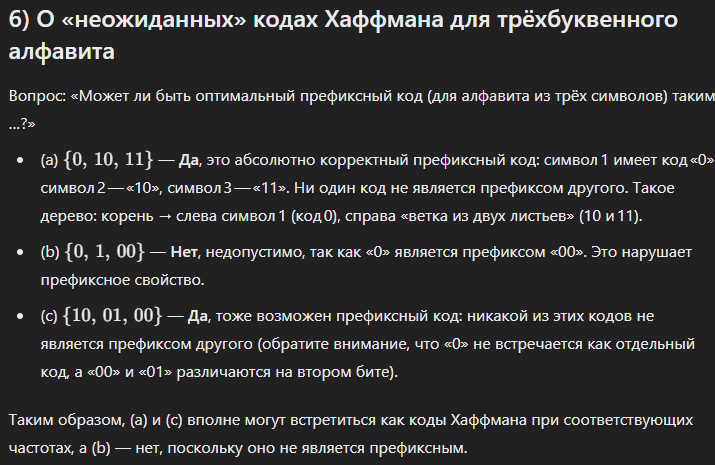


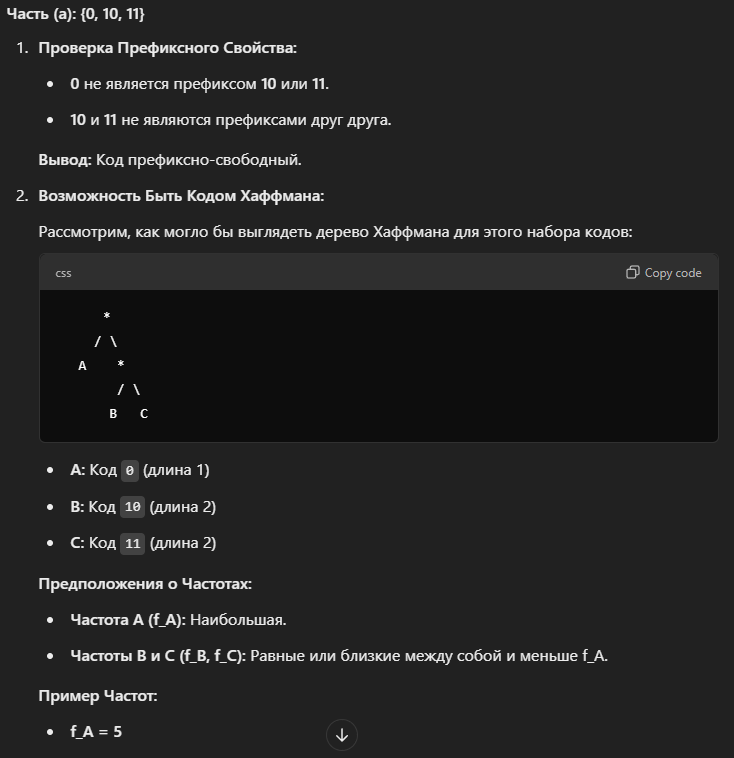
10.6) Может ли код Хаффмана для трехбуквенного алфавита оказаться таким (для каких-то частот):

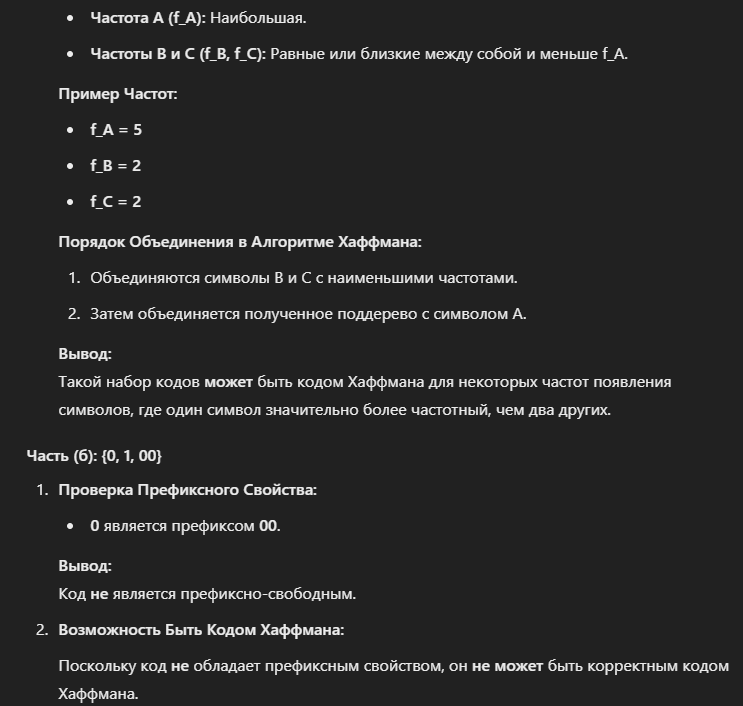
(а) {0, 10, 11};

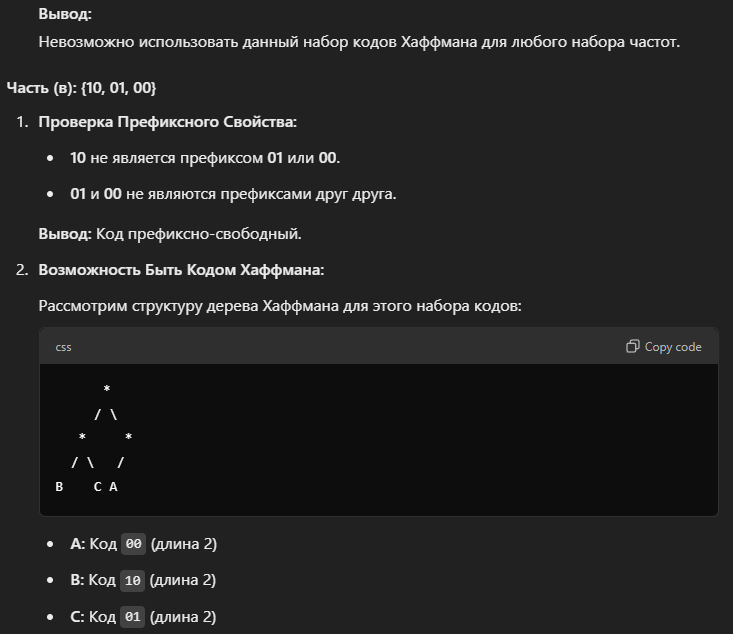
(б) {0, 1, 00};

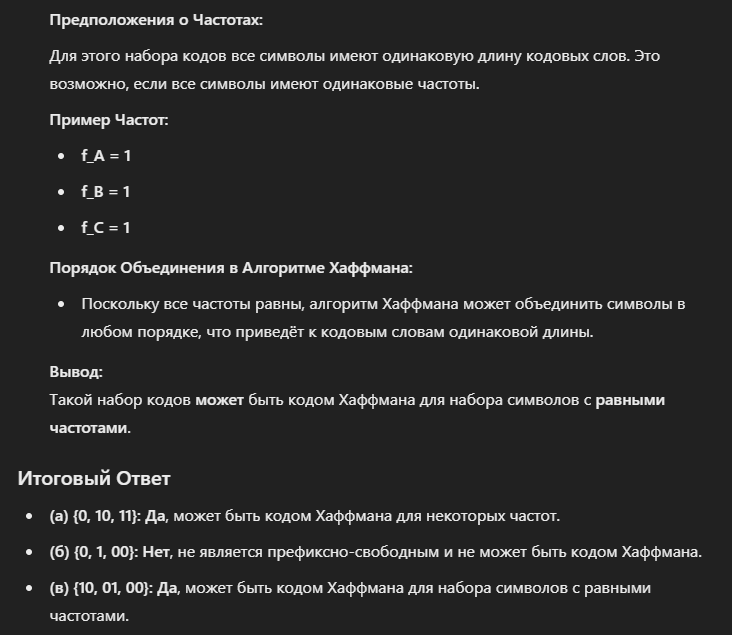
(c) {10, 01, 00}?



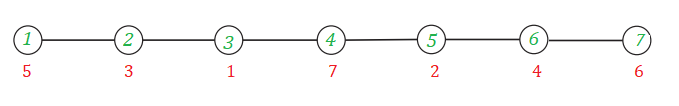




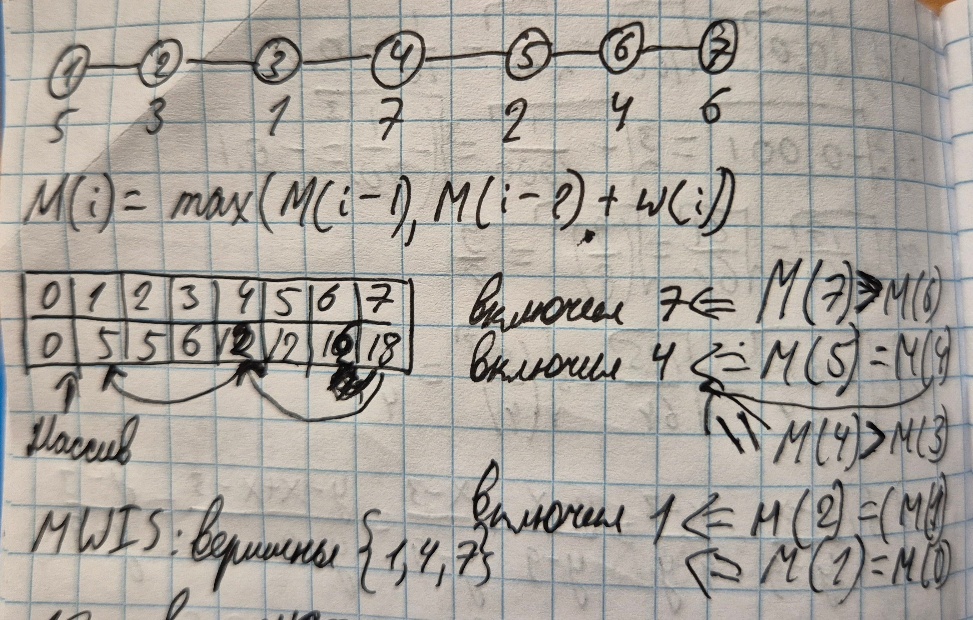




11.1) Рассмотрим входной путевой граф с вершинами {1, 2, 3, 4, 5, 6, 7}.



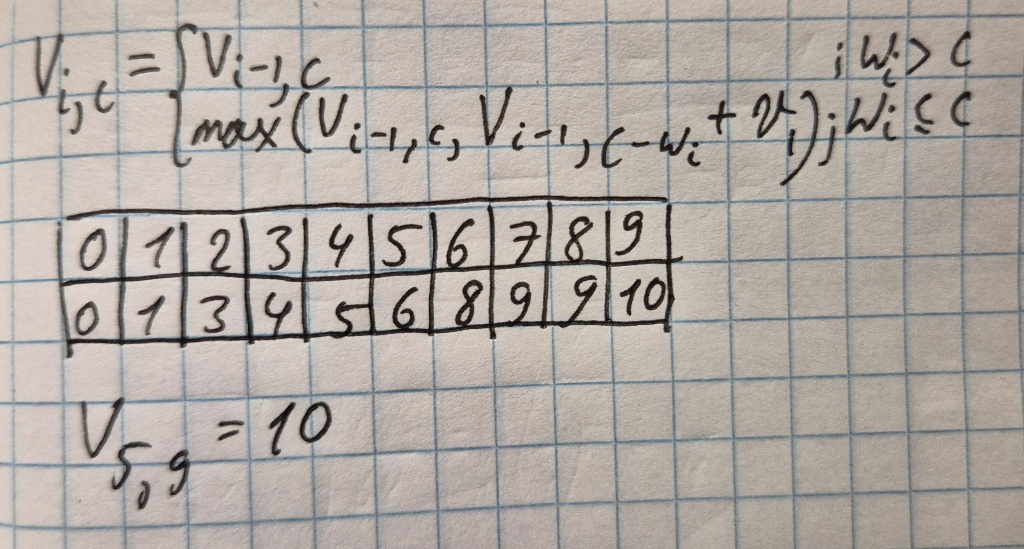
Красным цветом подписаны веса. Каковы конечные элементы массива в алгоритме WIS и какие вершины принадлежат независимому множеству максимального веса MWIS?



11.2) Рассмотрим пример задачи о рюкзаке с 5 предметами. Рюкзак емкостью *C*=9.

|  |  |  |
| --- | --- | --- |
| предмет | значение | размер |
| 1 | 1 | 1 |
| 2 | 2 | 3 |
| 3 | 3 | 2 |
| 4 | 4 | 5 |
| 5 | 5 | 4 |

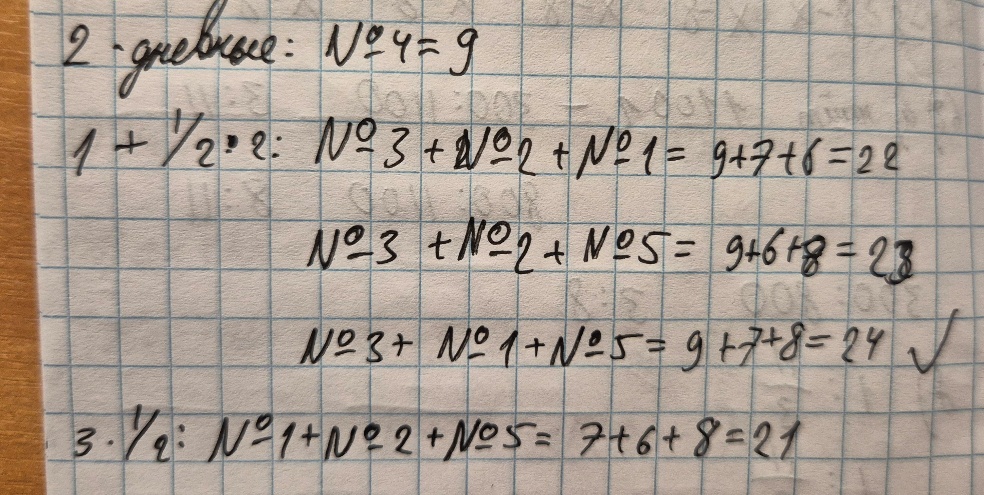
Какими являются окончательные элементы массива алгоритма Knapsack? Какие элементы принадлежат оптимальному решению?



11.3) Вы приехали в Санкт-Петербург на выходные. У вас 2 дня, мест, которые хочется посетить, слишком много. Побывать везде не получится, поэтому вы составляете список.

|  |  |  |  |
| --- | --- | --- | --- |
| № | достопримечательность | Время  (в днях) | оценка |
| 1 | Русский музей | 1/2 | 7 |
| 2 | Летний сад | 1/2 | 6 |
| 3 | Петергоф | 1 | 9 |
| 4 | Эрмитаж | 2 | 9 |
| 5 | Петропавл. крепость | 1/2 | 8 |

Для каждой достопримечательности вы указываете, сколько времени займет осмотр и насколько сильно вы хотите ее увидеть. Построить оптимальный туристический маршрут на основании этого списка.



* 1. Предположим, что в дискретной задаче о рюкзаке порядок сортировки по увеличению веса совпадает с порядком сортировки по уменьшению стоимости. Сформулируйте эффективный алгоритм для поиска оптимального решения этой разновидности задачи о рюкзаке.

