

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по практической работе №4
по дисциплине «Машинное обучение»

Студент гр. 1310

Комаров Д. Е.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2025

Задание 1

Постановка задачи

Дан набор данных, представленный в таблице 1.

Таблица 1 – Набор данных для 1 задания

Tid	Itemset
1	<i>ACD</i>
2	<i>BCD</i>
3	<i>AC</i>
4	<i>ABD</i>
5	<i>ABCD</i>
6	<i>BCD</i>

Найдите все минимальные генераторы для минимального уровня поддержки = 1. Множество является минимальным генератором, когда оно не имеет подмножеств с тем же уровнем поддержки.

Код программы

```
I={'A','B','C','D'}
D=[
    {'A','C','D'},
    {'B','C','D'},
    {'A','C'},
    {'A','B','D'},
    {'A','B','C','D'},
    {'B','C','D'}
]

def getSup(st,D):
    sup=0
    for d in D:
        if (st & d)==st:
            sup+=1
    return sup
def getNextC(L,I):
    C=set()
    for l in L:
        for i in I:
            if not(i in l):
                C=C|{(tuple(sorted(l|{i})))}
    C=[set(s) for s in C]
    return C

def apriori(D,I,minsup):
    F=[]
    C=[{i}for i in I]
    while C:
```

```

    L=[c for c in C if getSup(c,D)>=minsup]
    F.extend(L)
    C=getNextC(L,I)
return F

F=apriori(D,I,1)
rm=[]
for f in F:
    for ff in F:
        if f!=ff and getSup(f,D)==getSup(ff,D) and (f&ff)==f:
            rm.append(ff)
for r in rm:
    F.remove(r)
print(F)

```

Результат выполнения

В результате выполнения программы для минимального уровня поддержки 1 были получены следующие минимальные генераторы: $\{\text{'D'}\}$, $\{\text{'B'}\}$, $\{\text{'C'}\}$, $\{\text{'A'}\}$, $\{\text{'D'}, \text{'A'}\}$, $\{\text{'B'}, \text{'A'}\}$, $\{\text{'C'}, \text{'A'}\}$, $\{\text{'D'}, \text{'C'}\}$, $\{\text{'B'}, \text{'C'}\}$, $\{\text{'B'}, \text{'C'}, \text{'A'}\}$, $\{\text{'D'}, \text{'C'}, \text{'A'}\}$.

Задание 2

Постановка задачи

На рисунке 2 представлена решетка наборов и их частоты.

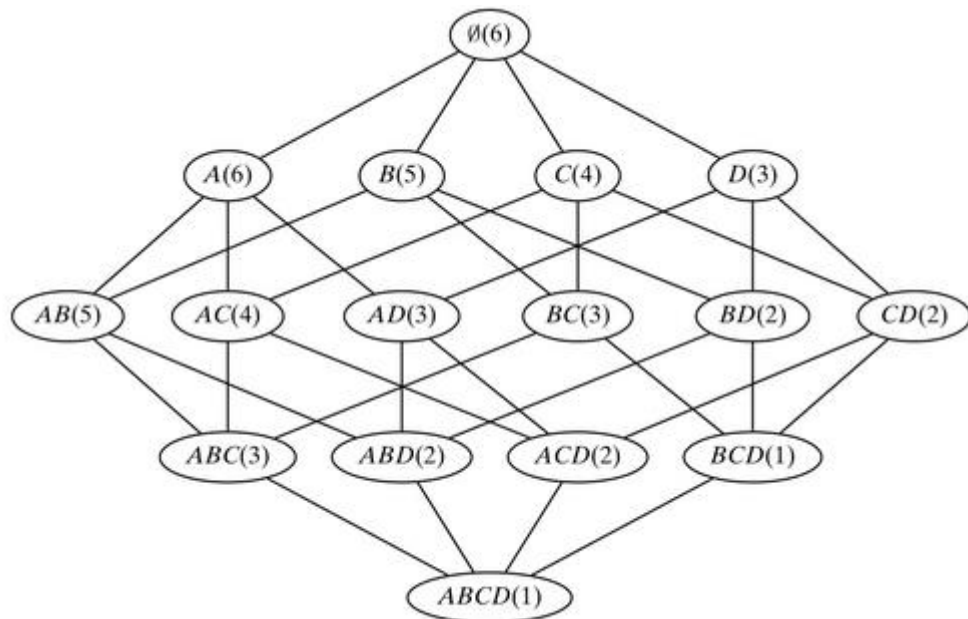


Рисунок 2 – Решетка наборов

Необходимо выполнить следующие задания:

- 1) Выписать список всех замкнутых наборов (closed itemsets).

Множество замкнутое, когда нет надмножеств с тем же уровнем поддержки.

2) Является ли набор BCD выводимым? Является ли набор ABCD выводимым? Какие границы их поддержки?

Результат выполнения

1) Замкнутые наборы: [$\{A, B, C, D\}$, $\{A, B, C\}$, $\{A, B, D\}$, $\{A, C, D\}$, $\{A, B\}$, $\{A, C\}$, $\{A, D\}$, $\{A\}$].

2) $LB(BCD) = \max(-5+3+2; -4+3+2; -3+2+2) = \max(0; 1; 1) = 1$;
 $UB(BCD) = \min(6-5-4-3+3+2+2; 3; 2; 2) = \min(1; 3; 2; 2) = 1$. Следовательно, так как $LB(BCD) = UB(BCD) = 1$, то набор BCD – выводимый.

$LB(ABCD) = \max(-6+18-19+8; -5+3+2; -4+3+2; -3+2+2; -3+3-1; -2+2+1; -2+2+2) = \max(1; 0; 1; 1; 1; 1; 1) = 1$;
 $UB(ABCD) = \min(6-5-4-3+2+2; 5-5-3-2+3+2+1; 4-4-3-2+3+2+1; 3-3-2-2+2+2+1; 2; 2; 1) = \min(1; 1; 1; 1; 2; 2; 1) = 1$. Следовательно, так как $LB(ABCD) = UB(ABCD) = 1$, то набор ABCD – выводимый.

Задание 3

Постановка задачи

1) Для алфавита $\{A, C, G, T\}$ посчитайте, сколько всего может быть разных последовательностей длины k.

2) Даны последовательности, представленные в таблице 2.

Таблица 2 – Последовательности для 3 задания

Id	Sequence
s_1	AATACAAGAAC
s_2	GTATGGTGAT
s_3	AACATGGCCAA
s_4	AAGCGTGGTCAA

Найдите все подпоследовательности с минимальным уровнем поддержки = 4.

Код программы

```
import re
I={'A','T','G','C'}
D=[
    'AATACAAGAAC',
    'GTATGGTGAT',
    'AACATGGCCAA',
    'AAGCGTGGTCAA'
]
```

```

def getSup(st,D):
    sup=0
    st="".join([sti+"*" for sti in st])
    for d in D:
        if re.search(st,d):
            sup+=1
    return sup

def getNextC(L,I):
    C=[]
    for l in L:
        for i in I:
            C.append(l+i)
    return C

def GSP(D,I,minsup):
    F=[]
    C=[i for i in I]
    while C:
        L=[c for c in C if getSup(c,D)>=minsup]
        F.extend(L)
        C=getNextC(L,I)
    return F

print(GSP(D,I,4))

```

Результат выполнения

- 1) Для алфавита {A,C,G,T} может быть 4^k последовательностей длиной k.
- 2) Подпоследовательности с минимальным уровнем поддержки = 4: ['A', 'G', 'T', 'AA', 'AG', 'AT', 'GA', 'TA', 'TG', 'AAT', 'AGA', 'ATA', 'ATG', 'GAA', 'TAA', 'TGA', 'ATGA']