

Множества

Занятие кружка по информатике ФИМЛИ 5

г. Долгопрудный, МО
Грицуляк РТ,
28 сентября 2016

Представление множеств

- Задача с мешком слов, на подумать в классе:
 - Есть множество слов $\{ w[0], w[1], w[2], \dots w[n-1] \}$
 - $n < 64$
 - Есть $f(\text{набор}) \rightarrow$ функция, которая работает непонятно как, но выдает нам на каждый набор.
 $\{w[i_1], w[i_2], w[i_3], \dots w[i_k]\} \rightarrow$ число. Например, это может быть хэш функция.
 - Известно, что количество документов не зависит от порядка слов и количества вхождений слов в список
 - 1) Найти набор на котором функция будет максимальна
 - Решение не должно зависеть от вида функции f .
 - 2) реализовать пересечения и объединения мешков слов

Мощность(1)

- множества подмножеств из $\{w[1], w[2], w[3], \dots, w[n]\}$ — булеан
- Мощность булеана (количество входящих в него элементов) 2^n
- Действительно, закодируем двоичное число с помощью элемента булеана следующим образом:
- Для подмножества $\{w[i_1], w[i_2], w[i_3], \dots, w[i_k]\}$
число $= 2^{i_1} + 2^{i_2} + 2^{i_3} + \dots + 2^{i_k}$
 - То есть для элемента множества i — если он есть в подмножестве — он есть и на i -й с конца позиции начиная с 0й.

```

int n;

vector<string> words;

cin >> n;

bitset<64> bits(0);

for(int i=0;i<n;i++) {

    string word;

    cin >> word;

    words.push_back(word);

}

int last_max=-1;

```

```

for(int j=0;j<(1<<n);j++)
{
    bitset<64> bitelem(j);

    bits=bitelem;

    vector<string> element;

    for(int k=0;k<n;k++){

        if(bitelem.test(k))

            element.push_back(words[k]);

    }

    int cur_max = f(element);

    if(cur_max>last_max)

        last_max=cur_max;

}

```

- Как будет выглядеть пересечение 2х множеств?
- Объединение?

```
Bitset<100> a(300); bitset<100> b(900);
```

```
c=a&b; d=a|b;
```

- Как будет выглядеть пересечение 2х множеств?

`a&b` `a and b`

$$a \cap b := a \wedge b$$

- Объединение

`a|b` `a or b`

$$a \cup b := a \vee b$$

Задачи

- Паскалистам — на паскале написать свой вариант для 30 слов (для 200 на ***)
.
- C++ - реализовать на `uint64_t` (допустим у нас меньше 64 слов), и для 200 слов на `uint64_t wrds[4]`
- Сравнить результаты на {«one», «two», «three», «four», «five», «six», «seven», «eight», «nine», «ten»}
- Попробовать «взломать» решения друг-друга (первые 2 решивших в пару итд)
- Решение = 700; взлом +100 ; решение «200» - 1500, +100

Задача на оптимальный перебор

- Допустим добавление и удаление множества занимает по 1
- Тогда перебор
- $sz|0 \rightarrow 1 \rightarrow 10 \rightarrow 11 \rightarrow 100 \rightarrow 101 \rightarrow 110 \rightarrow 111| = 11$
- хуже чем
 $sz|0 \rightarrow 1 \rightarrow 11 \rightarrow 111 \rightarrow 101 \rightarrow 100 \rightarrow 110 \rightarrow 10| = 7$
- Реализовать функцию оптимального перебора

Коды грея

- Описывают такие последовательности
- Можно ли доказать существование?

Доказательство

по индукции:

$G(1)$: последовательность $2^1 : \{0, 1\}$ — код грея
для $G(n+1)$ припишем 0 слева к $G(n)$ и
1 слева к $G^R(n)$ обратной последовательности
объединим: $G(n+1) = 0G(n) \cup 1G^R(n)$

proof : end ($G(n)$) = *begin* ($G^R(n)$)
дальше очевидно

Реализация

Задача

Реализация

```
for(uint32_t i=0;i<numeric_limits<uint32_t>::max();i++)  
{  
    bitset<32> out;  
    out = i ^ (i >> 1); // i xor (i shr 1) — pascal  
    cout << "i=" << i << ":" << out << endl;  
}
```

Задача

- Сложная: Написать декодер из кода грея — **в оригинальное представление** (на математику).
- проще: Модифицировать задачу с мешком слов, сравнить производительность, на вход функция принимает set (на олимпиадное программирование).

Вопросы и надеюсь ответы