На прошлой лекции мы познакомились с битовыми операциями. На этой лекции мы обсудим их применение. Битовые операции над целыми числами удобно использовать, если числа обозначают не количества каких-то предметов, а кодируют подмножества некоторого множества.

Рассмотрим некоторое множество элементов. Например, множество разноцветных бабочек на рис. 1.

Рис. 1. Множество

Можно взять какую-то часть его элементов, и у нас получится подмножество (см. рис. 2). Например, если взять красную и синюю бабочку — будет одно подмножество, красная, желтая и зеленая бабочки составляют другое подмножество. Подмножество может состоять из одной бабочки. Существуют два крайних случая подмножеств — это пустое подмножество, не содержащее элементов, (оно обозначается знаком $\varnothing$) и подмножество, совпадающее со всем множеством.

Рис. 2. Подмножества

Абстрагируемся от конкретных объектов (бабочек) и перейдем к числам. Пронумеруем бабочек, начиная с нуля (см. рис. 3). Тогда наши подмножества превратятся в подмножества чисел от 0 до 4. Например, подмножеству из красной и синей бабочки будет соответствовать подмножество $\{0, 3\}$.

Можно закодировать подмножества двоичными числами. Если подмножество содержит какой-то элемент, то в данном разряде будет стоять единица, иначе — ноль. Для примера на рис. 4 в подмножестве есть красная бабочка, имеющая номер 0, и синяя, имеющая номер 3. Значит, в нулевом и третьем

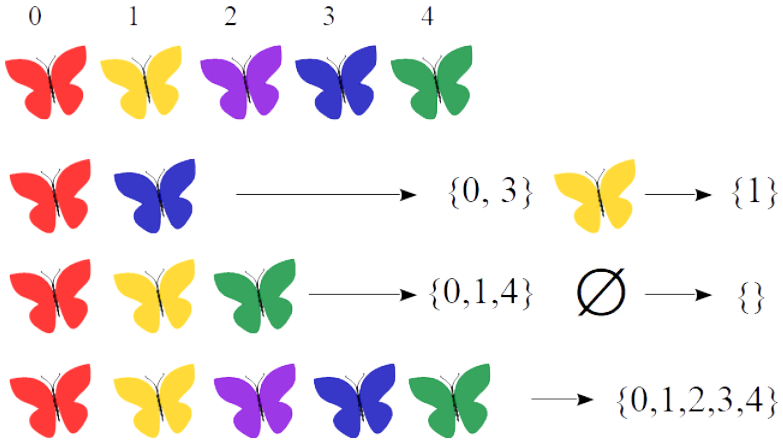

Рис. 3

разрядах будут единицы, в остальных — нули. Мы уже многократно использовали эту идею, когда работали с вектором used. В нем истинными значениями (единицами) отмечалось подмножество объектов, которые были уже использованы. Если множество достаточно маленькое, можно вместо вектора использовать одно число типа int.

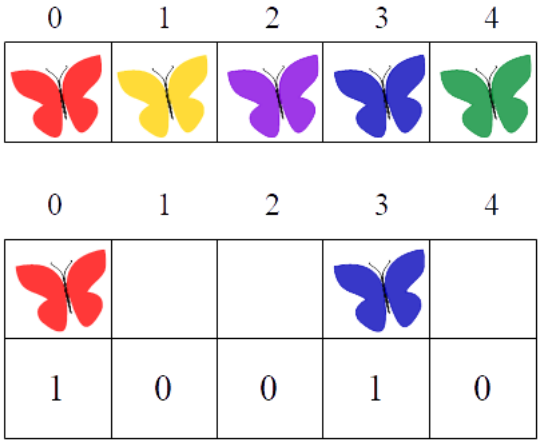

Рис. 4

Перевернём картинку (см. рис. 5), чтобы младшие разряды были в конце. Теперь номера бабочек соответствуют степеням двойки. Мы получили некоторое двоичное число. Подмножество из красной и синей бабочек соответствует числу 9.

Если мы добавим фиолетовую бабочку, то получим число 13 (см. рис. 6). В данном случае число 13 обозначает не количество предметов, а своими битами кодирует подмножество. Такие числа называют *битовыми масками*.

Упражнение 4.2.1 Какое число соответствует всему множеству из пяти бабочек? Дайте ответ в десятичной системе счисления.

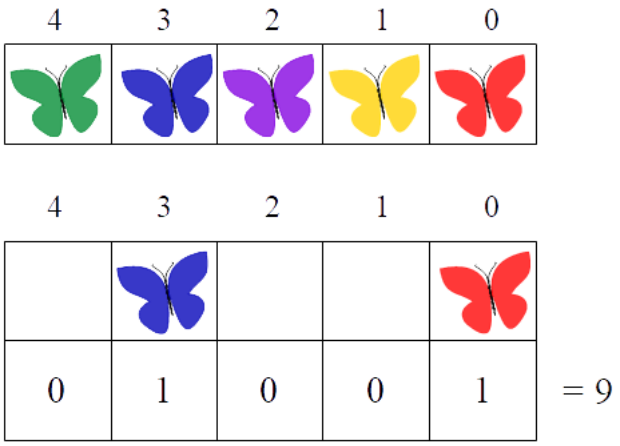

Рис. 5

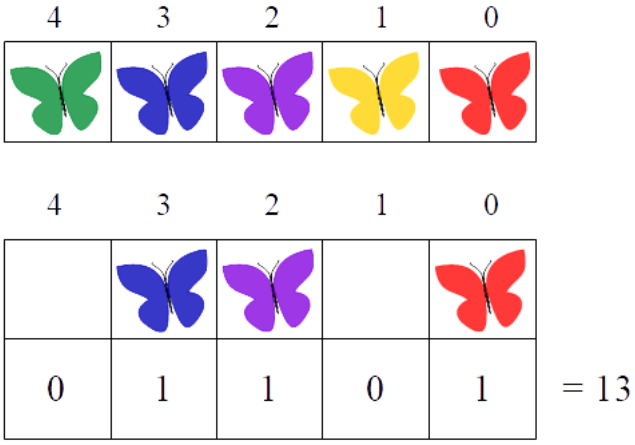

Рис. 6

Битовые операции соответствуют естественным операциям с подмножествами. Разберем пример с числами 13 и 14, который нам уже встречался (см. рис. 7). Операция AND оставляет единицы в тех позициях, где в обоих числах стояли единицы. При этом остаются только те бабочки, которые входят в оба подмножества. В данном случае — синяя и фиолетовая. На языке теории множеств эта операция называется *пересечением*.

Пересечение двух множеств — это новое множество, состоящее из элементов, которые входят в оба заданных множества (см. рис. 8). Операция AND с битовыми масками позволяет получить пересечение соответствующих подмножеств.

Аналогично рассмотрим операцию OR (см. рис. 9). Она оставляет единицы в тех позициях, в которых была хотя бы одна единица. Получается новое подмножество из бабочек, которые входят хотя бы в одно из заданных подмножеств.

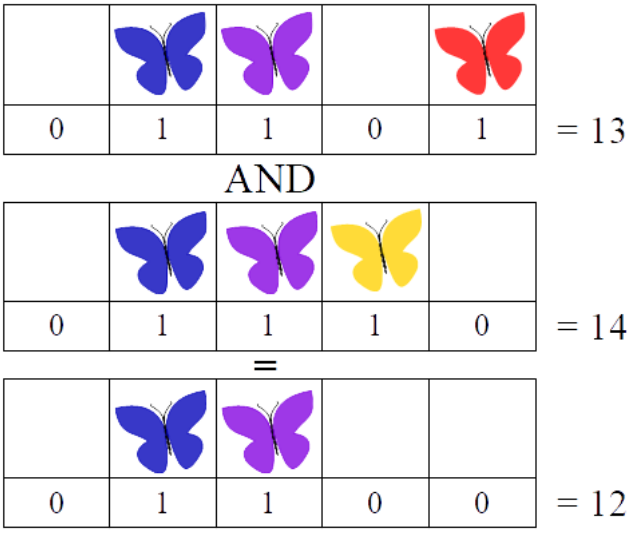

Рис. 7

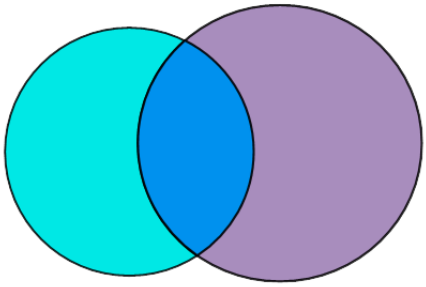

Рис. 8. Пересечение

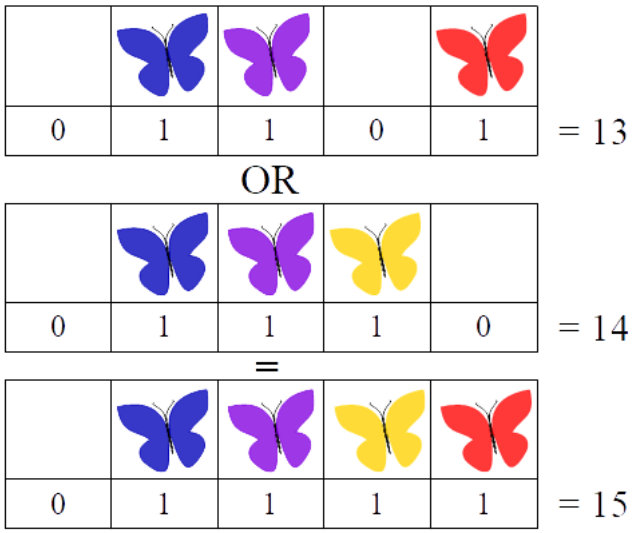

Рис. 9

Эта операция называется *объединением двух множеств* (см. рис. 10). Операция OR соответствует объединению.

На рис. 11 представлен результат операции XOR. При ее применении получится множество, составленное из элементов, входящих ровно в одно из

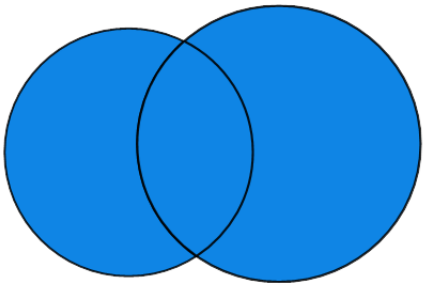

Рис. 10. Объединение

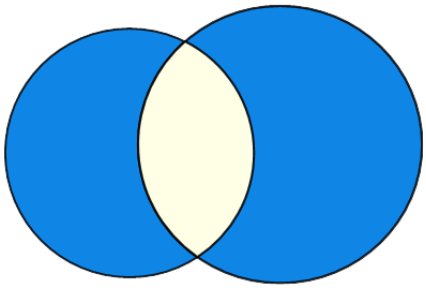

Рис. 11. XOR

исходных подмножеств. Иначе говоря, объединение подмножеств без их общей части.

Итак, мы разобрали операции AND, OR и XOR над битовыми масками и их связь с теорией множеств. Теперь попробуем решить задачу.

> Перебрать все возможные подмножества n-элементного множества чисел от 1 до n.

Например, для $n = 3$ будут следующие подмножества с разным количеством элементов:

$$\varnothing$$
$$\{1\}$$
$$\{2\}$$
$$\{3\}$$
$$\{1, 2\}$$
$$\{1, 3\}$$
$$\{2, 3\}$$
$$\{1, 2, 3\}$$

Эту задачу можно решить рекурсивным перебором, но можно поступить проще (см. рис. 12).

```
for (int mask = 0; mask < (1 << n); mask++)
{
    for (int i = 0; i < n; i++)
        if (mask & (1 << i))
            cout << i+1 << " ";
    cout << endl;
}
```

Рис. 12. Перебор подмножеств

В цикле перебираются значения переменной mask от 0 до $2^n - 1$:

for (**int** mask = 0; mask < (1 << n); mask++)

Так мы переберём все последовательности нулей и единиц длины n, соответствующие всем возможным подмножествам.

Вложенный цикл перебирает в маске все биты, соответствующие элементам в множестве:

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

Далее мы проверяем, есть ли в маске mask единица в i-м бите:

if (mask & (1 << i))

Рассмотрим число (1 << i). Оно равно 2^i, и его битовое представление содержит только одну единицу в i-й позиции. Если мы выполним операцию AND этого числа с маской, то результат будет равен этому числу, если в маске i-й бит равен единице, и получится 0, если в маске в i-м бите 0 (см. рис. 13). С точки зрения теории множеств, это пересечение множества, соответствующего маске, с множеством из одного элемента. if выполнится, если в маске i-й бит — единица. В этом случае число i содержится в подмножестве, и мы его выводим с прибавлением 1, чтобы перейти к 1-индексации:

cout << i + 1 << “ ”;

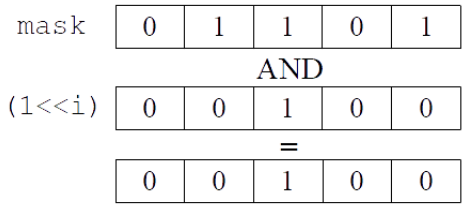

Рис. 13. Перебор подмножеств

Таким образом, мы перебрали все подмножества заданного множества при помощи битовых масок.