BIT? Xử lí bit trong C++

Tài liệu tóm tắt các phép toán bit, thao tác thường gặp, bitmask và ví dụ minh họa bằng C++.

Xử lí bit là thao tác trực tiếp trên từng bit 0/1 của số nguyên. Chủ đề này rất hay dùng khi cần tối ưu tốc độ, tiết kiệm bộ nhớ, kiểm tra chẵn lẻ, bật/tắt cờ, sinh tập con và giải các bài toán bitmask.

1. Các phép toán bit cơ bản

AND (&)

Bit chỉ là 1 khi cả hai bit đều là 1.

```
a & b
```

Ví dụ:

```
6 = 110
5 = 101
6 & 5 = 100 = 4
```

OR (|)

Bit là 1 khi ít nhất một bit là 1.

```
a | b
```

Ví dụ:

```
6 = 110
5 = 101
6 | 5 = 111 = 7
```

XOR (^)

Bit là 1 khi hai bit khác nhau.

```
a ^ b
```

Ví dụ:

```
6 = 110
5 = 101
6 ^ 5 = 011 = 3
```

NOT (~)

Đảo bit: 0 thành 1, 1 thành 0. Lưu ý trong C++ số nguyên dùng bù 2 nên kết quả có thể là số âm.

```
~a
```

Ví dụ:

```
~5
```

Dịch trái (<<)

Dịch bit sang trái k vị trí. Thường tương đương nhân với 2^k.

```
a << k
```

Ví dụ:

```
5 << 1 = 10
5 << 2 = 20
```

Dịch phải (>>)

Dịch bit sang phải k vị trí. Thường tương đương chia cho 2^k.

```
a >> k
```

Ví dụ:

```
20 >> 2 = 5
```

2. Các thao tác bit thường gặp

Giả sử đánh số bit từ phải sang trái, bắt đầu từ 0.

Lấy bit thứ k

```
(x >> k) & 1
```

```
int x = 13; // 1101
int k = 2;
cout << ((x >> k) & 1); // 1
```

Bật bit thứ k thành 1

```
x = x | (1 << k);
```

Tắt bit thứ k thành 0

```
x = x & ~(1 << k);
```

Đảo bit thứ k

```
x = x ^ (1 << k);
```

Kiểm tra bit thứ k có bật không

```
if (x & (1 << k)) {
    // bit k = 1
}
```

3. Một số mẹo hay

Kiểm tra số chẵn lẻ

```
if (x & 1) cout << "Le";
else cout << "Chan";
```

Bit cuối là 0 thì số chẵn, bit cuối là 1 thì số lẻ.

Nhân hoặc chia cho 2

```
x << 1   // x * 2
x >> 1   // x / 2
```

Xóa bit 1 thấp nhất

```
x = x & (x - 1);
```

Ví dụ: x = 12 (1100) thì x & (x - 1) = 1000 = 8. Ứng dụng nhiều trong đếm số bit 1.

Đếm số bit 1

```
int cnt = 0;
while (x) {
```

```
    x &= (x - 1);
    cnt++;
}
```

4. Bitmask trong C++

Bitmask là cách dùng một số nguyên để biểu diễn một tập hợp. Ví dụ, bit 0 bật nghĩa là có phần tử 0; bit 1 bật nghĩa là có phần tử 1; bit 2 bật nghĩa là có phần tử 2.

Ví dụ biểu diễn tập hợp

```
int mask = 0;

// them phan tu 2
mask |= (1 << 2);

// them phan tu 0
mask |= (1 << 0);
```

Khi đó mask = 101 (nhị phân) = 5, tức là tập hợp chứa {0, 2}.

5. Duyệt tất cả tập con bằng bitmask

Với n phần tử, ta duyệt các giá trị mask từ 0 đến (1 << n) - 1. Mỗi mask tương ứng với một tập con.

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

Ứng dụng: sinh tập con, quy hoạch động bitmask, các bài toán tổ hợp và biểu diễn trạng thái.

6. Hàm hỗ trợ có sẵn trong C++

Đếm số bit 1

```
__builtin_popcount(x);   // int
__builtin_popcountll(x); // long long
```

Đếm số bit 0 ở đầu

```
__builtin_clz(x);
```

Đếm số bit 0 ở cuối

```
__builtin_ctz(x);
```

Ví dụ:

```
cout << __builtin_popcount(13); // 3 vi 13 = 1101
```

7. Ví dụ đầy đủ

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int x = 10; // 1010
```

```
    cout << "Bit thu 1: " << ((x >> 1) & 1) << "\n";

    x |= (1 << 0); // bat bit 0
    cout << "Sau khi bat bit 0: " << x << "\n";

    x &= ~(1 << 3); // tat bit 3
    cout << "Sau khi tat bit 3: " << x << "\n";

    x ^= (1 << 1); // dao bit 1
    cout << "Sau khi dao bit 1: " << x << "\n";

    cout << "So bit 1: " << __builtin_popcount(x) << "\n";

    return 0;
}
```

8. Các bài toán bit hay gặp

- Kiểm tra chẵn lẻ
- Đếm số bit 1
- Bật, tắt, đảo bit
- Sinh tập con
- Tìm số xuất hiện lẻ lần bằng XOR
- DP bitmask
- Biểu diễn trạng thái

9. Ghi nhớ nhanh

```
x & (1 << k)      // ki□m tra bit k
x | (1 << k)      // bật bit k
x & ~(1 << k)     // t□t bit k
x ^ (1 << k)      // đ□o bit k
(x >> k) & 1      // l□y bit k
x & (x - 1)       // xóa bit 1 th□p nh□t
```

Tài liệu này phù hợp để ôn nhanh trước khi làm bài tập hoặc dùng làm ghi chú nhập môn về bit và bitmask trong C++.