Treinos Livres
Maratona de Programação

André Augusto
Humberto Longo
Paulo Cezar Pereira Costa

Instituto de Informática
Universidade Federal de Goiás

5 de abril de 2013

Aula 3
Estruturas de Dados (Continuação)

BitMask + BitSet

Estruturas de Dados (continuação)

Bitmask

- Assim como o *set* que foi citado anteriormente, existe outra forma de armazenar conjuntos de dados de forma leve e eficiente, e chamamos esta estrutura de dados de Máscara de Bits.
- Todo inteiro ou qualquer outro tipo de dado é armazenado internamente como uma sequência de bits.
- Para cada possível elemento de um conjunto, este elemento está ou não está no conjunto, o que pode ser representado também de forma binária com o uso de apenas um bit.

Bitmask

Para conseguirmos usar as bitmasks é necessário saber o básico de lógica booleana.

A	B	!A	A && B	A || B	A ^ B
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Figura : Topcoder

Estruturas de Dados (continuação)

Bitmask

A versão bit a bit destas operações funciona da mesma forma, mas ao invés de interpretar seus argumentos como verdadeiro ou falso, eles operam sobre cada bit dos argumentos. Logo, se A é 1010 e B é 1100, então:

- A & B = 1000
- A | B = 1110
- A ^ B = 0110
- ~A = 11110101 (o numero de 1's depende no número de bits do tipo de A).

Estruturas de Dados (continuação)

Bitmask

- Outros dois operadores que precisaremos são os *shift operators* $a << b$ e $a >> b$. O primeiro faz um *shift* de todos os bits em a para a esquerda por b posições. O segundo faz a mesma coisa mas o *shift* é pra direita.
- Para valores não negativos (que são os que estamos interessados), os novos bits que aparecem com os *shifts* são zeros.
- Podemos considerar *shifts* de b bits para a esquerda como uma multiplicação por 2^b e para a direira como divisão inteira por 2^b.
- Um dos usos mais comuns de *shifts* bit a bit são para acessar um bit em particular. Por exemplo, $1 << x$ é um número binário com o bit x setado e os outros não setados(bits são quase sempre contados como sendo o menos significante o mais a direita, o qual tem índice 0).

Estruturas de Dados (continuação)

Bitmask

- Em geral, usamos um inteiro para representar um conjunto em um domínio de até 32 valores (ou 64, usando um inteiro de 64 bits), com um bit setado representando um valor que está presente no conjunto e um bit não setado representando um valor que não está presente no conjunto.
- Sabendo disso, as seguintes operações são bem intuitivas, onde ALL_BITS é um número com 1's para todos os bits correspondendo aos elementos do domínio:
  - União de conjuntos: A | B
  - Intersecção de conjuntos: A & B
  - Subtração de conjuntos: A & ∼B
  - Negação de conjuntos: ALL_BITS ^ A
  - Setar um bit: A |= 1 « bit
  - Limpar um bit: A &= ∼(1 « bit)
  - Testar um bit: (A & 1 « bit) != 0

Estruturas de Dados (continuação)

Bitset

- Esta representação com inteiros entretanto é muito limitada, sendo esta limitação de apenas o maior inteiro representado pela linguagem, que normalmente é um inteiro de 64 bits.
- Para lidar com conjuntos maiores, existe uma implementação na STL que satisfaz esta necessidade com alguma perda de performance chamada *bitset*.

Estruturas de Dados (continuação)

Bitset

- Operadores usados nas bitmasks também podem ser usados com bitsets:
  - &, |, ^ , «, », ∼
- Além destes, vários outras funcionalidades são implementadas

Estruturas de Dados (continuação)

Bitset

- Acesso a bits pelo operador [ ]
- set() – Seta os bits do bitset
- reset() – Reseta os bits do bitset
- flip() – Inverte os bits do bitset
- to_ulong() – Retorna o bitset convertido para um unsigned long
- to_string() – Retorna o bitset convertido para uma string
- count() – Retorna o número de bits ativos
- size() – Retorna o número de bits do bitset
- test(pos) – Retorna se o bit na posição *pos* está setado

Leituras Recomendadas

- http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=bitManipulation
- http://www.codechef.com/wiki/tutorial-bitwise-operations
- http://www.comp.nus.edu.sg/~stevenha/visualization/bitmask.html
- http://inf.ufg.br/~paulocosta/tap/material/bitmask.pdf