**Bitset**

bitset用法大全：

复杂度O(n/w)

定义：bitset<maxn>bs;

构造：

bitset(unsigned long val): 设为 val 的二进制形式。

bitset(const string& str): 设为 01 串 str。

遍历:

b[0]表示二进制最低位，例如:bs:01001那么b[0] = 1;

可以使用cin;cout输入输出

成员函数：

bs.count():返回bs中为true的数量；

bs.size():返回bitset 的大小。

test(pos): 它和 vector 中的 at() 的作用是一样的，和 [] 运算符的区别就是越界检查。

any(): 若存在某一位是 true 则返回 true，否则返回 false。

none(): 若所有位都是 false 则返回 true，否则返回 false。

all():C++11，若所有位都是 true 则返回 true，否则返回 false。

set(): 将整个 bitset 设置成 true。

set(pos, val = true): 将某一位设置成 true/false。

reset(): 将整个 bitset 设置成 false。

reset(pos): 将某一位设置成 false。相当于 set(pos, false)。

flip(): 翻转每一位。（0\leftrightarrow1，相当于异或一个全是 1 的 bitset）

flip(pos): 翻转某一位。

to\_string(): 返回转换成的字符串表达。

to\_ulong(): 返回转换成的 unsigned long 表达（long 在 NT 及 32 位 POSIX 系统下与 int 一样，在 64 位 POSIX 下与 long long 一样）。

to\_ullong():C++11，返回转换成的 unsigned long long 表达。

应用：

（1）bitset代替vis数组实现算法常数优化

（2）bitset优化背包，用二进制表示可以实现空间与时间优化：

例子：

bitset<maxn> bt;

int main()

{

int n,a,sum = 0;

scanf("%d",&n);

bt[0] = 1;

rep(i,0,n){

scanf("%d",&a);

bt |= bt<<a;

sum += a;

}

rep(j,(sum+1)/2,sum+1) if(bt[j]) return printf("%d",j),0;

return 0;

}