**Bit Magic**

*Important points:*

* To check if a number ‘N’ is a power of 2 then the trick is that only 1 bit in it would be set.
* Pow (2, x) [means 2^x] may be written in bit logic as **1 << x**
* To “turn off” the rightmost set bit in number ‘n’: **n & (n-1)**
* To check if Kth bit from right is set or not in number ‘n’:
* if (**n & (1 << (k-1))**) is zero, then the Kth bit from right is 0(not set)
* else the Kth bit from right is 1(set)
* To retain only the Kth set bit from right (and make all other bits 0) in number ‘n’:

**n & ~(n-1)**

**Arrays**

*Important points:*

* KADANE’s Algorithm
* Maximum normal Sub-Array sum => Minimum circular Sub-Array sum
* Moore’s Algorithm