Important array question

→ Compute x^n (Binary Exponentiation)

Suppose

$x^n = 3^5$

5 → 101

3^4 3^1

ans = $3^4 * 3^1$

$x = 3^1 \cdot 3^1$

$x^2 = 3^2 \cdot 3^2$

$x^4 = 3^4 \cdot 3^4$

$x^8 = 3^8$

dec $n = \log_2 n + 1$ → Binary form Numbers

$O(\log n)$

dry run : binform = n, ans = 1, x → 3^1, 3^2, 3^3

101

(1) { ans = ans * [x] = 3^1
x = x * x = 9

(2) { x = x^2 = 81

(3) { ans = ans * x
= 3 * 81 = 243
x = x^2 = $(81)^2$

Pseudocode

for +ve

```
while(BF>0){
  if(BF%2==1){
    ans *= x
  }
  x* = x/x^2
  BF%=2;
}
```

O(log n)

For -ve

$(3)^{-5} = (\frac{1}{3})^5$

```
if(n<0){
  x = 1/x
  binf = -binf
}
```

Adding Corner cases :-

```
if(n==0) return 1
if(x==0) return 0
if(x==1) return 1
if(x==-1) return 88(n%2==0) 1
if(x==-1 && n%2==0) return 1
```

→ **Stock By & Sell**

7	1	5	3	6	4

most be buy day — Selling ⑥

Most Diff Days
Min P — Max P
Buy — Sell — Diff
Prof

Imagine each day as selling day and calculate the profit for each day and find the day which gives the maximum profit.

MaxProfit = M.P = 0

BestBuy → index (0 to idx-1) / min value

T.C

```
for(i=1; i<n; i++){
    if(price[i] > bestbuy){
        m.p = max(mp, price[i] - bestbuy)
    }
    bestbuy = min(BB, price[i])
}
```

O(n) linear one

m.p gives the best selling day by giving maximum profit.

bestbuy gives the best buying day