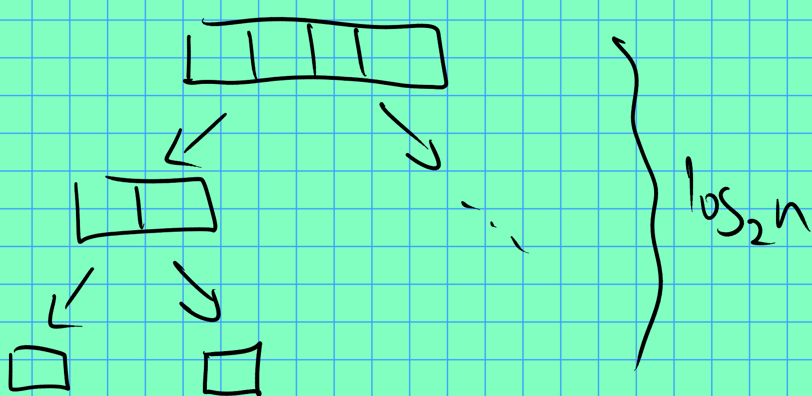
```
mergeSort(int* A, int n, int* Aux)
{
    if (n<2) return;
    int mid = n/2;
    mergeSort(A, mid, Aux);   //Sort A[0,...mid-1]
    mergeSort(A+mid, n-mid, Aux);   //sort A[mid,...n-1]
    merge(A, mid, A+mid, n-mid, Aux);
    //now copy Aux -> A ...
    for(i=0; i<n; i++) A[i]=Aux[i];
}
```



Extended GCD algorithm.

Fact: if $d = \gcd(a,b)$, then $\exists\ x, y \in \mathbb{Z}$

s.t. $d = ax + by$.

How to find x & y?

$$\gcd(12,18) = 6 = 12\cdot(-1) + 18\cdot 1$$

$$x = -1, \quad y = 1.$$

```
int xgcd(int a, int b, int& x, int& y) {
               outputs
        if (b==0) {
            x = 1;
            y = 0;        //a = a·1 + b·0
                                 x     y
            return a;
        }

        int xx, yy, q = a/b, r = a%b;

        int d = xgcd(b, r, xx, yy);

      // d = b·xx + r·yy

    // Note : a = qb + r

    // goal: find x,y s.t. d = ax+by.

    // but a=qb+r => r = a-qb

    // So, d = b·xx + (a-qb)·yy

    //        = a·yy + b(xx-qyy)

    x = yy;

    y = xx - qyy;

    return d;
}
```