

```

// P(a,i,j) is all products of a[i..(0..j)]
// P*(a,i,j) is all P(a,0..i,j)
// -----
// if 0 <= n <= actual length of a
// and INT_MIN <= P*(a,n-1,n-1) <= INT_MAX
// then msp(a,n) is the maximum of P*(a,n-1,n-1)
int msp(int a[],int n)
{
    int m = 1; // pre loop1
    int i = 0;
    while (i<n) { // take loop1
        // 0 <= n <= actual length of a
        // INT_MIN <= P(a,n-1,n-1) <= INT_MAX
        // m = max of P*(a,i-1,n-1)
        // i <= n

        int j = i;
        m = maxFromHere(j, n, m, a);
        i++;

    } // exit loop1
    return m;
}

** pre loop1
// 0 <= n <= actual length of a
// INT_MIN <= P(a,n-1,n-1) <= INT_MAX
int m = 1;
int i = 0;
// 0 <= n <= actual length of a (A)
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX (B)
// m = max of P*(a,i-1,n-1) (1,2,P*,empty range)
// i <= n (A,2)

** take loop1
// 0 <= n <= actual length of a
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX
// m = max of P*(a,i-1,n-1)
// i <= n
|- i<n
int j = i;
m = maxFromHere(j, n, m, a);
i++;
// 0 <= n <= actual length of a (A)
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX (B)
// m = max of P*(a,i-1,n-1) and P(a, i, j-1) (C,3)
// i <= n (D,1)

** exit loop1
// 0 <= n <= actual length of a
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX
// m = max of P*(a,i-1,n-1) and P(a, i, j-1)
// i <= n
~ i<n
return m;
// m = max of P*(a,n-1,n-1) (C,D,1)
// THUS msp(a,n) = max of P*(a,n-1,n-1) (2)

```

```

// maxFromHere(j, n, m, a) returns m, which is the max of all products of a[j..
(0..n-1)] and m
// -----
// if 0 <= n <= actual length of a
// and INT_MIN <= maxFromHere(j, n, m, a) <= INT_MAX
// maxFromHere(j, n, m, a) returns m, which is the max of all products of a[j..
(0..n-1)] and m
int maxFromHere(int i, int n, int m, int a[]) {
    int p = 1;
    int j = i;
    while (j < n) {
        // take loop2
        // 0 <= n <= actual length of a
        // INT_MIN <= P(a,n-1,n-1) <= INT_MAX
        // m = max of P*(a,i-1,n-1) and P(a,i,j-1)
        // i <= n
        // p = product of a[i..j-1]
        // j <= n
        p *= a[j];
        if (p > m)
            m = p;
        j++;
    }
    return m
}

** pre loop2
// 0 <= n <= actual length of a
// INT_MIN <= P(a,n-1,n-1) <= INT_MAX
// i <= n
int p = 1;
int j = i;
// 0 <= n <= actual length of a
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX
// m = max of P*(a,i-1,n-1)
// i <= n
(A)
(B)
(1,2,P*,empty range)
(A,2)

** take loop2
// 0 <= n <= actual length of a
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX
// m = max of P*(a,i-1,n-1)
// i <= n
// p = product of a[i..j-1]
// j <= n
|- j < n
p *= a[j];
if (p > m)
    m = p;
j++;
// 0 <= n <= actual length of a
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX
// m = max of P*(a,i-1,n-1) and P(a, i, j-1)
// i <= n
// p = product of a[i..j-1]
// j <= n
(A)
(B)
(C,3)
(D,1)

** exit loop2
// 0 <= n <= actual length of a
// INT_MIN <= P*(a,n-1,n-1) <= INT_MAX

```

```

// m = max of P*(a,i-1,n-1) and P(a, i, j-1)
// i <= n
// p = product of a[i..j-1]
// j <= n
~ j < n
return m;
// m = max of P*(a,i,n-1) (C,D,1)
// THUS maxFromHere(j, n, m, a) = max of P*(a,i,n-1) (2)

```