Algorithm 1 Skyline-Search

Input: $D = \{(x_i, y_i) | i = 1, 2, ..., n\}$: the points in the plane;
Output: D': the remain points in the plane that can not be conquered;

if $D = \Phi$ **then**
    return $D' = \Phi$;
else if D has only one point (x_0, y_0) **then**
    return $D' = (x_0, y_0)$;
else
    sort all the points in D based on the value of x and we have $x'_1 < x'_2 < ... < x'_n$';
    initial $L = [(x'_1, y'_1), ..., (x'_n, y'_n)]$, mid = $[\frac{n}{2}]$;
    initial left=$\{(x'_1, y'_1), ..., (x'_{mid}, y'_{mid})\}$;
    initial right=$\{(x'_{mid+1}, y'_{mid+1}), ..., (x'_n, y'_n)\}$;
    l=Skyline-Search(left);
    r=Skyline-Search(right);
    find the point p with the biggest y in r;
    for all point i in the l **do**
        if y of $p > y$ of i **then**
            remove i from l;
        end if
    end for
    $D' = l \cup r$;
end if
return D'