Algorithm 2DBOUNDEDLP($H,\vec{c},m_1,m_2$)

Input. A linear program ($H \cup \{m_1,m_2\},\vec{c}$), where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.

[redacted]

Output. If ($H \cup \{m_1,m_2\},\vec{c}$) is infeasible, then this fact is reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

Let v_0 be the corner of C_0.
Let $h_1,\dots,h_n$ be the half-planes of H.
for $i \leftarrow 1$ **to** n
    do if $v_{i-1} \in h_i$
        then $v_i \leftarrow v_{i-1}$
        else $v_i \leftarrow$the point p on ℓ_i that maximizes $f_{\vec{c}}(p)$, subject to the constraints in H_{i-1}.
            if p does not exist
                then Report that the linear program is infeasible and quit.
return v_n

Algorithm 2DRANDOMIZEDBOUNDEDLP($H,\vec{c},m_1,m_2$)

Input. A linear program ($H \cup \{m_1,m_2\},\vec{c}$), where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.

Output. If ($H \cup \{m_1,m_2\},\vec{c}$) is infeasible, then this fact is reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

Let v_0 be the corner of C_0.
Compute a *random* permutation $h_1,\dots,h_n$ of the half-planes by calling RANDOMPERMUTATION($H[1\cdots n]$).
for $i \leftarrow 1$ **to** n
    do if $v_{i-1} \in h_i$
        then $v_i \leftarrow v_{i-1}$
        else $v_i \leftarrow$the point p on ℓ_i that maximizes $f_{\vec{c}}(p)$, subject to the constraints in H_{i-1}.
            if p does not exist
                then Report that the linear program is infeasible and quit.
return v_n