P1

Prompt:

Given a rectangular slab of size W x H and a set of rectangular plates of sizes W_1 x H_1, W_2 x H_2, ... W_n x H_n, design an algorithm that runs in time polynomial in n, W, and H and that outputs the minimum possible waste area.

Algorithm:

```
Input: Slab dimensions W and H and a set of plates P.
Output: The minimum possible waste area.

Let t be an (W + 1) x (H + 1) table
Let every entry in t = −1

maxCut(W, H, P):
    If t[W][H] != −1 then
        Return t[W][H]
    EndIf

    Let C be a copy of P
    For each plate p in P do
        If p.w = W and p.h = H then
            Set t[W][H] = W * H
            Return t[W][H]
        Else if W < p.w or H < p.h then
            Remove p from C
        EndIf
    EndFor

    Let vertCutMax = 0
    Let horizCutMax = 0

    For each plate p in C do
        If p.w != W then
            Let vertCut = maxCut(W − p.w, H, C) + maxCut(p.w, H, C)
            If vertCut > vertCutMax then
                Set vertCutMax = vertCut
            EndIf
        EndIf

        If p.h != H then
            Let horizCut = maxCut(W, H − p.h, C) + maxCut(W, p.h, C)
            If horizCut > horizCutMax then
                Set horizCutMax = horizCut
            EndIf
```

```
        EndIf
    EndFor

    Let maxCut = max{vertCutMax, horizCutMax}
    t[W][H] = maxCut
    return t[W][H]

Let maxCut = maxCut(W, H, P)
Output (W * H) - maxCut
```

Claim:

The algorithm terminates in time polynomial in n, W, and H.

Proof:

The algorithm iterates over each plate p in P at most 2n times. In the first of these loops, the work performed is constant in time, making the dominant runtime factor the n loop iterations. In the second loop, the work is divided into 4 subproblems: 2 subproblems of size W-p.w and p.w and 2 sub problems of size H-p.h and p.h, respectively. In each case, when a solution to a subproblem is found it is stored in the (W+1) x (H+1) table making future checks of that same sized subproblem accessible in constant time. Therefore, the runtime of the problem is bounded by W * H * 2n since there are W * H entries that must be checked and for each entry as many as 2n checks against the n plates in P. Since 2 is a constant factor, we ignore that and can say that the runtime of the problem is bounded by W * H * n. Since each subproblem is monotonically smaller in size than the calling problem, since each loop iterates at most n times, and since the problems are bounded on the low end (a plate p is removed from P if it's larger than W x H), the algorithm terminates. Thus, that algorithm terminates in time polynomial in n, W, and H.

Claim:

The algorithm outputs the minimum possible waste area.

Proof:

Let OPT(W, H) denote the maximum usable area cut from a slab with dimensions W x H into plates from a set P = {W_1 x H_1, W_2 x H_2, ... W_n x H_n}.

Let v(W, H) be a function that takes the slab dimensions W x H and returns the area of that slab if there is a plate p of size W x H in P and returns 0 otherwise.

$$\text{v(W, H)} = \begin{cases} \text{W * H} & \text{if p.w = W and p.h = H for any p in P} \\ 0 & \text{otherwise} \end{cases}$$

Case 1: If a slab with dimensions W x H is not cut, then the maximum usable area of that slab is W * H if there there is a plate p in P such that p.w = W and p.h = H. Otherwise, if no such plate p exists, then the maximum usable area of a slab with dimensions W x H is 0. This is given by the defined v(W, H) function.

$$\text{OPT(W, H) = v(W, H)}$$

Case 2: If a slab with dimensions W x H is cut vertically, then the cut that yields the maximum usable area for the slab with dimesions W x H will be the cut yielding two slabs with dimensions (W - p.w, H) and (p.w, H) for some plate p in P such that the sum of the maximum usable area of the two cut slabs is the maximum over all plates p in P.

$$\text{OPT(W, H) = max\{OPT(W - p.w, H) + OPT(p.w, H)\} for all p in P}$$

Case 3: If a slab with dimensions W x H is cut horizontally, then the cut that yields the maximum usable area for the slab with dimesions W x H will be the cut yielding two slabs with dimensions (W, H - p.h) and (W, p.h) for some plate p in P such that the sum of the maximum usable area of the two cut slabs is the maximum over all plates p in P.

$$\text{OPT(W, H) = max\{OPT(W, H - p.h) + OPT(W, p.h)\} for all p in P}$$

Thus, the recurrence relation that maximizes the usable area of the slab:

$$\text{OPT(W, H)} = \max \begin{cases} \text{v(W, H)} & \text{If not cut} \\ \text{max\{OPT(W - p.w, H) + OPT(p.w, H)\} for all p in P} & \text{If cut vertically} \\ \text{max\{OPT(W, H - p.h) + OPT(W, p.h)\} for all p in P} & \text{If cut horizontally} \end{cases}$$

To find the minimal wasted area, simply subtract the maximum usable area given by OPT(W, H) from the total area of the slab, W * H. □