Решение: "Meet in the Middle"

Задачата се решава с техника, наречена **"Meet in the Middle"**. Това е ефективен подход за проблеми, които могат да бъдат разделени на две части и където директното изброяване на всички възможности е прекалено скъпо.

Стратегия

Целта е да преброим броя на пътищата от горния ляв ъгъл `(1, 1)` до долния десен ъгъл `(N, M)`, при които сумата на елементите по пътя е делима на `P`. Поради ограничението на времето за изпълнение не можем да генерираме всички пътища директно.

Вместо това разделяме пътя на две части:

1. Генерираме всички пътища от `(1, 1)` до средата (диагонала).
2. Генерираме всички пътища от `(N, M)` до средата.
3. Съчетаваме резултатите, като намираме тези комбинации, при които общата сума е делима на `P`.

Основна идея: Честотен брояч по модул

За да преброим валидните пътища ефективно, поддържаме **честотен брояч по модул `P`** за всяка клетка на диагонала:

- За всяка клетка `(x, y)` по диагонала:
  - Записваме броя на сумите по модул `P` за пътищата от `(1, 1)` до `(x, y)`.
  - Аналогично, записваме броя на сумите по модул `P` за пътищата от `(N, M)` до `(x, y)`.

Стъпки

1. **Генериране на пътища от `(1, 1)` до диагонала:**
   Използваме DFS, за да намерим всички пътища до дадена клетка по диагонала. За всяка клетка поддържаме масив от размер `P`, който съдържа честотата на всяка сума по модул `P`.

2. **Генериране на пътища от `(N, M)` до диагонала:**
   Аналогично, използваме DFS, за да генерираме всички пътища, които започват от `(N, M)` и достигат клетка по диагонала, като отново поддържаме честотен брояч по модул `P`.

3. **Комбиниране на резултатите:**
   За всяка клетка на диагонала:

   - Нека `fwd_mod[x][y]` е честотният брояч на сумите по модул за пътища от `(1, 1)` до `(x, y)`.
   - Нека `bwd_mod[x][y]` е честотният брояч на сумите по модул за пътища от `(N, M)` до `(x, y)`.
   - Съчетаваме двата брояча, като броим всички комбинации, при които общата сума е делима на `P`.

Сложност

- **Времева сложност:**
  Генерирането на пътища от `(1, 1)` и `(N, M)` включва около `2^(N-1)` и `2^(M-1)` пътища за всяка половина. Комбинирането на резултатите е пропорционално на броя клетки по диагонала и `P`, тъй като трябва да фиксираме всеки възможен остатък.

  Общо: `O((N + M) * (P + 2^N + 2^M))`.

- **Паметова сложност:**
  `O(P * (N + M))`, тъй като поддържаме честотни броячи за всяка клетка на диагонала.