**Динамично оптимиране**

 Първото нещо, което трябва да забележим, е че можем да третираме **N**-те рафта като един голям рафт, като разстоянието между 2 позиции в един рафт го броим за 1 а между два рафта за **∞**. (всъщност, всяко число над 1000 би свършило работа, тъй като по условие има не повече от 1000 позиции общо по всички рафтове).

 Нека означим общият брой позиции с **P**, с **B[i]** да обозначим активността на **i**-тата позиция, независимо от рафт, в реда, в който са въведени.

За да получим добро решение на задачата е нужно да използваме 2D динамично оптимиране, като за всяко възможно разстояние между две колби

1 ≤ **W** ≤ 1000, проверим каква е оптималната наредба (ако такава е възможна) 1 колба, 2 колби, и т.н. до M колби и 1 позиция, 2 позиции и т.н. до P позиции. Нека **best[<брой\_колби>][<брой\_позиции>]** ни е най-доброто решение за даден брой колби и даден брой позиции. За всяко **W**, **best[1][1]** **= A[1]** (използваме индексиране от 1. Най-доброто решение за 1 позиция и една колба е просто да сложим тази колба на единствената възможна позиция) и след това **best[1][k] = min(best[1][k-1], B[k])** за k ≥ 1. За всяка следваща бройка колби **m**, първо линейно намираме първата позиция **x**, за която тя може да бъде осъществена, и **best[m][x] = best[m-1][x-W]** ако в реда, на който е x-тата свободна позиция, преди x има поне W свободни места, или **best[m][x] = best[m-1][x-W’]** в противен случай, където x-W’ е позицията на последния елемент от предходния ред.

**Оптимално решение**

За да получим оптимално решение на задачата, можем да използваме горното решение, с модификацията, че ще търсим максималната възможна стойност на W с двоично търсене, като трябва да забележим, че ако подредба не е възможна за някое W, то със сигурност няма да е възможна за кое да е X > W.