

1 3.1 Высокая пирамида

Дано N кубиков. Требуется определить каким количеством способов можно выстроить из этих кубиков пирамиду. Каждый вышележащий слой пирамиды должен быть не больше нижележащего. $N \leq 200$.

2 Описание алгоритма

- Выделяем таблицу $N + 1 \times N + 1$ чисел. Заполняем нулевой, первый столбик и первую строку единицами. Первое число двумерного массива $[i]$ - это максимальное число, которое может находиться в нижнем слое, второе число $[j]$ - всего кубиков, использующиеся при построении пирамиды.
- Каждый набор в табличке состоит из 2-х поднаборов:
 - $arr[i][j - 1]$ - количество наборов, которые могут быть "навершием" пирамидки.
 - $arr[i - 1][j]$ - количество наборов с меньшим размером основания.
- Выполнением программы происходит рекурсивным вызовом от последней клетки $func(arr[N][N])$. В функции происходит присваивание $a[i][j] = a[i][j - 1] + arr[i - 1][j]$ и возвращение этого значения.

3 Доказательство корректности работы

Корректность работы следует из единственности представления набора в клетке (см. пункт описание алгоритма).

4 Время работы и доп. память

- Время работы $O(n^2)$, где n - количество кубиков.
- Доп. память $O(n^2)$.

5 Доказательство времени работы

Произойдет не более n^2 вызовов функции (размер) таблицы \implies время работы $O(n^2)$. На программу выделяется таблица $n \times n$ и глубина рекурсии не более, чем $2 \times n \implies$ дополнительная память равна $O(n^2)$