

Сходинки

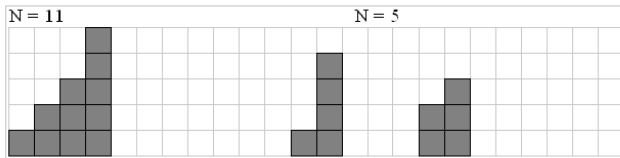
Лясковець Сергій

Київський національний університет
імені Тараса Шевченка
фізичний факультет

7 квітня 2018 р.

Постановка задачі

У маленького хлопчика є набір з N кубиків ($5 \leq N \leq 500$). З цих кубиків можна скласти різні сходи. Сходи мають ступені різного розміру, які прямують у порядку зростання цього розміру (зверніть особливу увагу на те, що сходи не можуть мати дві однакові ступені). Кожна драбина повинна мати мінімум дві сходинки, і кожна сходинка повинна складатися мінімум з одного кубика. На малюнку наведено приклади сходів для $N = 11$ і $N = 5$:



Знайдіть число Q різних сходів, які маленький хлопчик може побудувати рівно з N кубиків.

- Input format: число N
- Output format: Число Q

Створюємо матрицю, елементи якої це кількість можливих сходів. Заповнюємо матрицю по числу n вихідних кубиків і кількості s стовпців. Номер стрічки це кількість кубиків, номер стовпчика - кількість сходинок. У клітинку матриці $m(n, 1)$ ставимо 1. Решта клітини заповнюємо по формулі $f(n, s) = f(n - s, s) + f(n - s, s - 1)$. Тут перший доданок відображає випадок, коли в першому стовпці більше одного кубика, а друге - коли рівно один. Обчислюємо значення, поки не дійдемо до нульового. Якщо $n = N$, то, обчислюючи значення за формулою (2), одночасно знаходимо їх суму, що і буде відповіддю.

Реалізація

```
int  ftotal( int  all_cubes )
{
    int    res = 0;
    int    numb_comb[501][501]; //масив кубиків та стрічок, в стрічках - кількість кубиків, а стовпчиках - номер сходинки

    for( int  cubes = 1; cubes <= all_cubes; cubes++ )
    {
        numb_comb[cubes][1]  =  1;
        int    cols = 2;

        //рахуємо кількість комбінацій за допомогоюї рекурсивної формули
        while  (numb_comb[cubes][cols] = numb_comb[cubes - cols][cols] + numb_comb[cubes - cols][cols - 1])
        {
            if( cubes  ==  all_cubes )
            {
                res  +=  numb_comb[cubes][cols];
            }

            ++cols;
        }
    }

    return  res;
}
```

Рис. 1 : Використаний код

Реалізація

```
int    main()

{ //вводимо значення кількості кибикиів та виводимо результат
    // unsigned int start_time =  clock();
    int    N_all_cubes;

    cin    >>  N_all_cubes;

    cout   <<  ftotal( N_all_cubes )<<endl;
    // unsigned int end_time = clock(); // конечное время
    // unsigned int search_time = end_time - start_time; // искомое
    //  cout<<search_time<<endl;
}
```

Рис. 2 : Використаний код