

### 0.0.1 Оценка трудоемкости алгоритмов

Используется C-подобная модель оценки трудоемкости.

Трудоемкость операций:

- 1) трудоемкость операций: +, -, =, + =, - =, <, > ==, ++ равна 1;
- 2) трудоемкость операций: \*, /, % равна 2;
- 3) трудоемкость операции доступа к элементу памяти: [...] равна 3.

Трудоемкость смены ячеек памяти местами будем считать 9, так как производится 3 обращения к памяти.

Цикл будет оцениваться о фактически выполненным операциям из перечня выше.

Условный оператор if будет фактически оценен как сумма стоимости операций в условии и трудоемкости различных ветвей (в лучшем случае и в худшем случае). Стоимость условного перехода из условия в одну из ветвей решения полагается равной 0.

### 0.0.2 Классический алгоритм умножения матриц

Умножаются 2 матрицы M1(M, N), M2(N, Q). Ниже приведена формула расчета сложности в соответствии с моделью оценки трудоемкости.

Итоговая формула:

$$O(M, N, Q) = 8 * M * N * Q + 4 * M * N + 4 * M + 2 \quad (1)$$

### 0.0.3 Алгоритм Винограда

Умножаются 2 матрицы M1(M, N), M2(N, Q)

- 1) цикл 1:

$$2 + M(2 + 3 + \frac{N}{2} * (3 + 10)) \quad (2)$$

- 2) цикл 2:

$$2 + Q(2 + 3 + \frac{N}{2} * (3 + 10)) \quad (3)$$

- 3) цикл 3:

$$2 + M(2 + 2 + Q(2 + 7 + 3 + \frac{N}{2} * (3 + 20))) \quad (4)$$

Условие четности/нечетности:

$$1 + \begin{cases} 2 + M(2 + 2 + Q(2 + 10)), N \text{ нечетное (худший вариант)} \\ 0, \text{ иначе (лучший вариант)} \end{cases} \quad (5)$$

Итоговая формула получается из суммы 2, 3, 4, 5

#### 0.0.4 Оптимизированный алгоритм Винограда

Итоговая формула получается из суммы  $??$ ,  $??$ ,  $??$ ,  $??$