③

a) $O(n^3)$ for trivial algorithm

$$T(n) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} c =$$

Псевдокод

```
def trivial_mul(a, b)
  c[n][n]
  for (i, i<n, i++){
    for (j, j<n, j++){
      c[i][j] = 0
      for (k, k<n, k++){
        c[i][j] += a[i][k] *
                 * b[k][j]
      }
    }
  }
```

$$= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} nc =$$

$$= \sum_{i=0}^{n-1} n^2 c = n^3 c = O(n^3)$$

b) $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} p_5 + p_4 - p_2 + p_6 & p_1 + p_2 \\ p_3 + p_4 & p_5 + p_1 - p_3 - p_7 \end{bmatrix}$

Осю ці p_n – це 7 множень, отже:

$$T(n) = \overset{a}{7}\, T\left(\frac{n}{\underset{b}{2}}\right) + O(n^2)$$

a – к-ть підзадач; $O(n^2)$ – додавання та віднімання матриць

$$T(n) = O\left(n^{\log_2 7}\right)$$

Доведено

3

a) Complexity for median-search and k-smallest

Best case $- O(n)$
Worst case $- O(n^2)$

1. Вибір pivot $- O(1)$
2. Обхід кожного ел. для порівняння з pivot $- O(n)$
3. Рекурсивні виклики для двох підмасивів

$$\underbrace{a_0, a_1, a_3 \ldots}_{len = k} \text{pivot}_{a_k} \ldots \underbrace{a_{n-2}\; a_{n-1}\; a_n}_{len = n-k-1}$$

$$T(n) = T(k) + O(n)$$

В кращому випадку pivot ділить масив на дві ± рівні частини, тоді

$$T(n) = \underset{a}{1}\, T\left(\frac{n}{\underset{b}{2}}\right) + O(n)$$

Тоді, використаю основну теорему:

$$n^{\log_b a} = n^{\log_2 1} = 1 \Rightarrow$$

$$T(n) = \Omega(n)$$

Доведено.

В найгіршому випадку, якщо масив відсортований і півот це найбільший або найменший елемент, то:

$$T(n) = T(n-1) + O(n)$$ ← арифм. прогресія

. . . . . не розписую рекурсію

$$T(n) = O(n^2)$$

b) Очікується, що розмір масиву для кожного рекурсивного виклику зменшується на однаковий фактор.

Номери фаз $\overline{0, n}$

У фазі k розмір масиву не перевищує $n\left(\frac{3}{4}\right)^k$

Остання фаза $\lceil \log_{\frac{4}{3}} n \rceil$

Візьму X_k – випадкова величина

Робота у фазі k $< X_k c \cdot n \left(\frac{3}{4}\right)^k$

M – загальна робота

$$M \le \sum_{k=0}^{\lceil \log_{4/3} n \rceil} \left(X_k \cdot cn \left(\frac{3}{4}\right)^k \right) =$$

$$= cn \sum_{k=0}^{\lceil \log_{4/3} n \rceil} \left(X_k \left(\frac{3}{4}\right)^k \right) \leftarrow \heartsuit$$

$$E(M) \le E(\heartsuit)$$

Використовуючи властивості мат. очікування:

$$E(M) = cn \sum_{k=0}^{\lceil \log_{4/3} n \rceil} E[X_k] \left(\frac{3}{4}\right)^k$$

За означенням:

$$E[X_k] = \sum_{i=0}^{\infty} i \cdot P(X_k = i)$$

значення ймовірність цього значення

Всі вибори pivot є незалежні, тому

$$P(X_k = i) = \left(\frac{1}{2}\right)^i$$

$$E(X_k) = \sum_{i=0}^{\infty} i\, P(X_k = i) \le \sum_{i=1}^{\infty} \frac{i}{2^i} = 2$$

ряд збіжний

Тоді $E(M) \le 2cn \sum_{k=0}^{\infty} \left(\frac{3}{4}\right)^k = 8cn$

$E(M) = 8cn$

отже $E(M) = O(n)$

доведено

c) З лекції верхня оцінка часу

$$T(n) \le \sum_{k=0}^{\infty} X_k \cdot cn \left(\frac{3}{4}\right)^k$$

тоді використаємо принцип індукції, одразу за послідовністю

① База індукції $n = 1$

$$T(1) \le \sum_{k=0}^{\infty} X_k \cdot c \cdot 1 \cdot \left(\frac{3}{4}\right)^k$$

геом прогресія

$S_n = \frac{b_1}{1-q}$

$$\sum_{k=0}^{\infty} X_k \cdot c \cdot \left(\frac{3}{4}\right)^k = c \cdot X_0 \cdot \frac{1}{1-\frac{3}{4}} = 4c \cdot X$$

Отже, $T(1) \leq 4C \cdot X_0$ базовий

(2) Припустимо, що виконується для $i \leq n-1$, тоді викон для $i=n$

$$T(i) \leq \sum_{k=0}^{\infty} X_k \cdot ci \left(\frac{3}{4}\right)^k$$

$$T(n) = T\left(\frac{3n}{4}\right) + cn$$

підставимо

$$T\left(\frac{3n}{4}\right)$$