ЛЕКЦИЯ 6. АЛГОРИТМЫ СОРТИРОВКИ

```
ПРИМЕР:  a = [3, 7, 8, 2, 4, 6, 1, 5]
```

КВАДРАТИЧНЫЕ АЛГОРИТМЫ

СОРТИРОВКА ОБМЕНОМ ИЛИ «ПУЗЫРЬКОВАЯ»

```
for i in range(n):
    for j in range(n-i-1):
        if a[j]>a[j+1]:
            a[j], a[j+1] = a[j+1], a[j]
```

СОРТИРОВКА ВЫБОРОМ

```
for i in range(n-1):
    m = a[i]; ind = i
    for j in range(i+1,n):
        if a[j]<m:
            m = a[j]; ind = j
    a[ind] = a[i]; a[i] = m
```

СОРТИРОВКА ВСТАВКОЙ

```
for i in range(1,n):
    x = a[i]; j = i
    while (j>0) and (a[j-1]>x):
        a[j] = a[j-1]
        j -= 1
    a[j] = x
```

СЛОЖНОСТЬ АЛГОРИТМОВ

«о» малое и «О» большое:

$$f(x) = o(g(x)) \Leftrightarrow \frac{f(x)}{g(x)} \to 0$$

$$f(x) = O(g(x)) \Leftrightarrow \frac{f(x)}{g(x)} < C$$

Сложность:

- в худшем случае;
- в лучшем случае;
- в среднем (по умолчанию).

Сложность всех рассмотренных алгоритмов – $O(N^2)$

Доказательство: $f(N) = N + (N-1) + \ldots + 1 = \frac{(N+1)N}{2} = O(N^2)$

УСТОЙЧИВОСТЬ

ОПРЕДЕЛНИЕ. Сортировка называется устойчивой, если она не меняется относительный порядок элементов, имеющих одинаковые ключи.

Выбором – не устойчивая. Обменом – устойчивая. Вставкой – устойчивая.

БЫСТРЫЕ АЛГОРИТМЫ

СОРТИРОВКА СЛИЯНИЕМ (ДЖОН ФОН НЕЙМАН, 1945)

```
b = [0]*n
def merge(p,q,r):
    global a,b
    i = p; j = q+1
    for k in range(p, r+1):
        if (i<=q) and (j<=r) and (a[i]<a[j]) or (j>r):
            b[k] = a[i]; i += 1
        else:
            b[k] = a[j]; j += 1
    a[p:r+1] = b[p:r+1]
def sort(p,r):
    if p<r:
        q = (p+r) // 2
        sort(p,q); sort(q+1,r);
        merge(p,q,r);
sort(0,n-1)
```

QUICKSORT (ЧАРЛЬЗ ХОАР, 1962)

```
def sort(p,r):
    if p>=r:
        return
    global a
    i, j = p, r
    pivot = a[random.randint(p,r)]
    while i<=j:
        while a[i]<pivot: i += 1
        while a[j]>pivot: j -= 1
        if i<=j:
            a[i], a[j] = a[j], a[i]
            i += 1; j -= 1
    sort(p,j)
    sort(i,r)
sort(0,n-1)
```

СЛОЖНОСТЬ БЫСТРЫХ АЛГОРИТМОВ

Сложность сортировки слиянием – $O(N\log(N))$

Доказательство:

$$f(N) \le 2f\left(\frac{N}{2}\right) + N$$

Пусть $N = 2^k$. Тогда

$$f(2^k) = 2f(2^{k-1}) + 2^k = 2(2f(2^{k-2}) + 2^{k-1}) + 2^k = 2^2 f(2^{k-2}) + 2^k + 2^k = 2^3 f(2^{k-3}) + 2^k + 2^k + 2^k = \ldots$$

$$\ldots = 2^k + k2^k = N + N\log(N) = O(N\log(N))$$

Сложность быстрой сортировки: в худшем случае - $O(N^2)$, в среднем - $O(N\log(N))$.

ДЛЯ СРАВНЕНИЯ:

За сколько времени эти алгоритмы отсортируют 1 млрд. чисел?

Пусть компьютер выполняет 1 млрд. оп/сек

Квадратичный: $\frac{10^{18}}{10^9} = 10^9 c = 277777 час = 11574 дней = 31 год$

Быстрый: $\frac{30 \cdot 10^9}{10^9} = 30c$