



Алгоритмы

Анализ по алгоритмам - Корень

Размерность задачи \sim кол-во входных данных

Временная сложность \sim зависимость от времени

Пространственная сложность \sim зависимость от выделяемой памяти

Асимптотика

линейно $\rightarrow C_1 \sim O(N)$
 квадратично $\rightarrow C_1 + C_2 N^2 \sim O(N^2)$
 логарифмически $\rightarrow C_1 + C_2 \log N \sim O(\log N)$

Поиск пути в ориентированном графе

Поиск i-го ранжированного элемента $N \cdot \log N$ элементов

$1 \sim N \cdot \log N \Rightarrow N \sim \frac{1}{\log N} \Rightarrow I = \log N \Rightarrow$ сложность $O(\log N)$
 3-й шаг поиска

Сортировка пузырьком



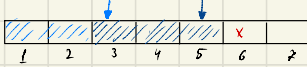
```
for (i=0; i < N; i++) {
    bool check = false;
    for (j=i; j < N-1; j++)
        if (array[j] > array[j+1])
        {
            check = true;
            int tmp = array[j];
            array[j] = array[j+1];
            array[j+1] = tmp;
        }
    if (!check) break;
}
```

Критерий "свернуто" для идеальной,
 где отсортированы массивы

[1 5 2 10]

[1 2 3 4]

1 5 10



2 100

400

2 400

low = 0

high = n-1

middle = $\frac{low + high}{2}$

if $x < a[middle]$ $x > a[middle]$
 high = middle - 1 low = middle

while low \leq high

{ return middle;
 return -1

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |