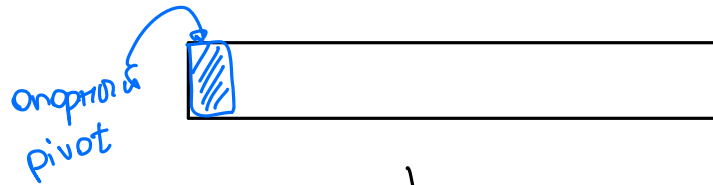


Quick Sort (быстрая сортировка)

① Пересчет $O(n \log n)$

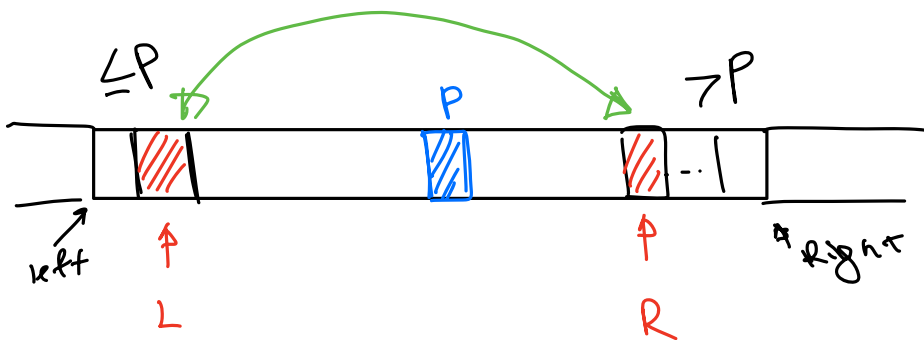
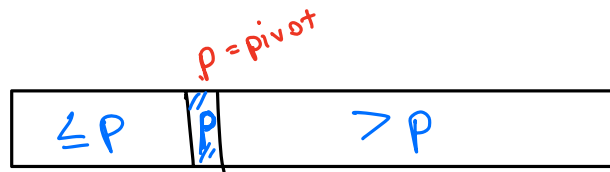
Partition
Quick Sort

② Partition loop.



pivot = arr[0]
arr[2-3]

↓ partition



Ans:

Partition(array, left, right)

{ pivot = arr[left]

L = left

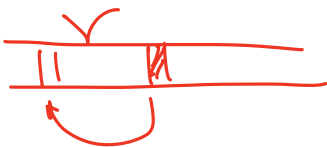
R = right

while L < R:

{

while L < right and arr[L] ≤ pivot:
 L = L + 1

while R > left and arr[R] > pivot



↳ $R = R - 1$

$t = arr[L]$
 $arr[L] = arr[R]$
 $arr[R] = t$

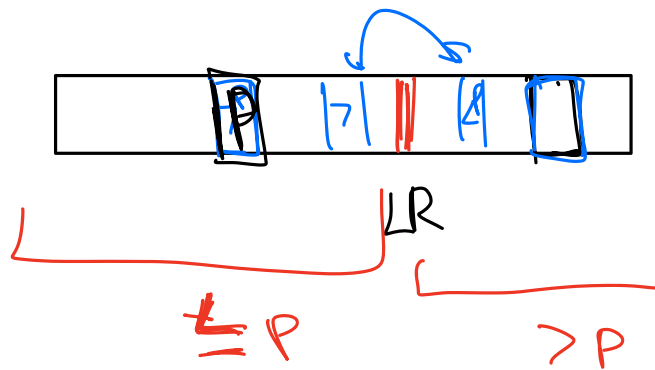
|| $swap(arr[L], arr[R])$

}
 }
 return L

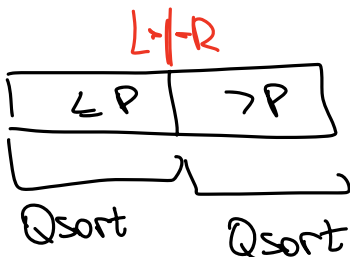
Время:

$$T(N) = \underline{O(N)}$$

"Прокси"



III Qsort



Qsort(arr, left, right)

если $right - left \leq 3$:
 ↳ return

$p = \text{Partition}(arr, left, right)$

Qsort(arr, left, p)

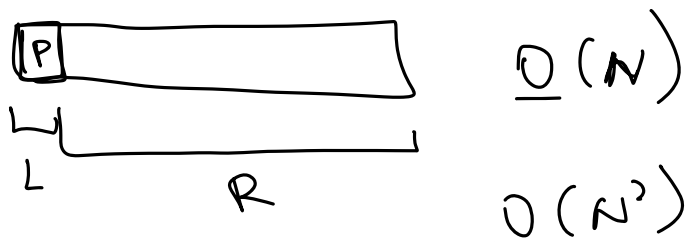
Qsort(arr, p+1, right)

$\log N$

Асимп: $\underline{O(N \log N)}$

$M(N) = \underline{O(1)}$

Прокси элемент:



IV Базов pivot

array
array // → on copy.

arr [rand(0...n)] \Rightarrow Teorema
 $O(n \log n)$

arr [$\frac{n}{2}$]

Хримо!

Medusa medusa

Сравнение:

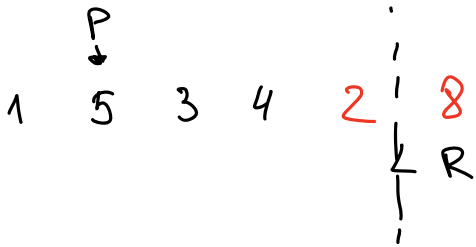
Bubble

Билбором

⑤ Verbrauch $N < 50$

⑤ MergeSort \rightarrow $\underbrace{O(n \log n)}$

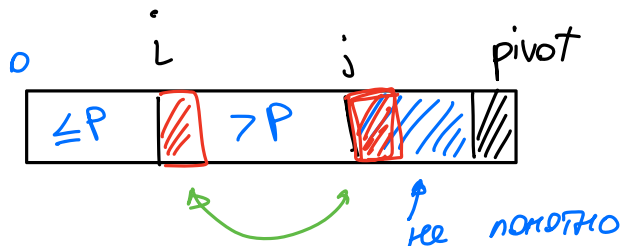
\textcircled{V} $Q_{\text{sent}} \rightarrow \text{docs}$
 $\textcircled{-} \quad \underline{O(N^2)}$
 moner.



⑤ Partition Anonymo



Une page sur Partition



§. pivot → 6 koruny.

Partition (arr, left, right)

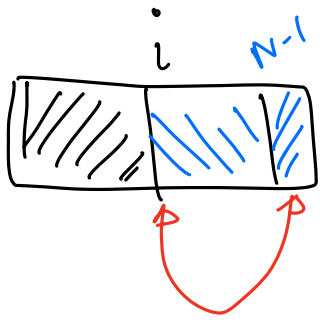
$pivot = arr[right]$

$i = \text{left}$

```

for j = left... right-1
{

```



```

    if arr[j] < pivot
        swap(arr[j], arr[i])
        i = i + 1
}
swap(arr[right], arr[i])
return i + 1

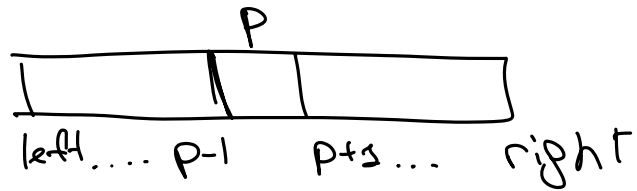
```

```

}
swap(arr[right], arr[i])
return i + 1

```

Qsort



VI) Сортировка Хоугемана (не эффективна)

a)

arr = [0 1 1 0 1 0 1 ...]

count_0 = 0

count_1 = 0

for i = 0... arr.size

if arr[i] == 0

count_0 += 1

count[0] += 1

else

count_1 += 1

count[1]

for i = 0... count_0

print(0)

for i = 0... count_1

print(1)

O(N)

count[arr[i]]
0
1

8) *not a max/min*

```

arr
max = arr[0]
for i = 0... arr.size
    because max < arr[i]
        max = arr[i]

```

count = [0... 0] // по умолчанию max+1

```

for i = 0... arr.size
    count[arr[i]] += 1

```

```

for i = 0... count.size:
    for j = 0... count[i].size
        print(i)

```

Значение 0 max [0... max]

[min ... max]

→ count[arr[i] - min]