به نام خدا

روش تقسیم و غلبه

مسأله مرتب سازی

Insertion sort بازگشتی

merge sort

زمان الگوریتم $T(n)$

merge-sort(A, p, r)

$\theta(1)$ {
1. if $p \geq r$ then return
2. $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$
}

$T(\frac{n}{2})$ 3. merge-sort(A, p, q)

$T(\frac{n}{2})$ 4. merge-sort(A, q+1, r)

$\theta(n)$ 5. merge(A, p, q, r)

فاز تقسیم ↑   فاز ترکیب ↑

زمان الگوریتم $= T(n) = 2T(\frac{n}{2}) + \theta(1) + \theta(n) = 2T(\frac{n}{2}) + \theta(n) \in \theta(n\lg n)$

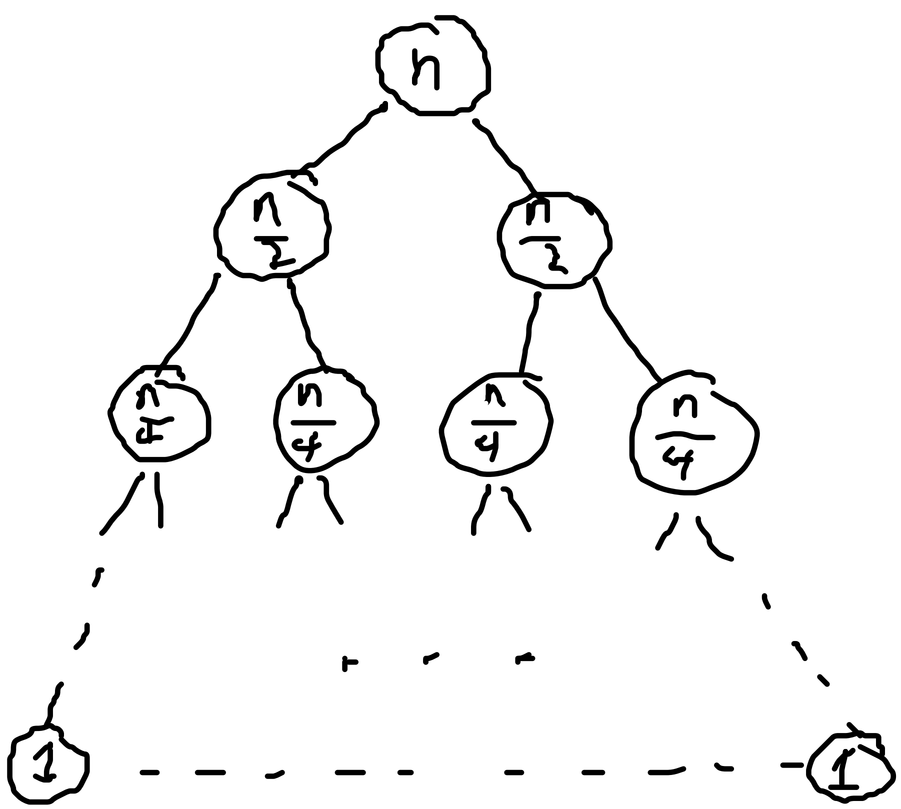
زمان الگوریتم : در هر حالت بررسی می‌شود

بهترین حالت : (کمترین زمان)

بدترین حالت : (بیشترین زمان)

حالت متوسط : (میانگین)

فراخوانی ها بازگشت



در مورد الگوریتم merge_sort :

درخت باینری کامل

زمان الگوریتم $= T(n) = \Theta(n \lg n)$

merge_sort

بدترین حالت : $O(n \lg n)$

بهترین حالت : $\Omega(n \lg n)$

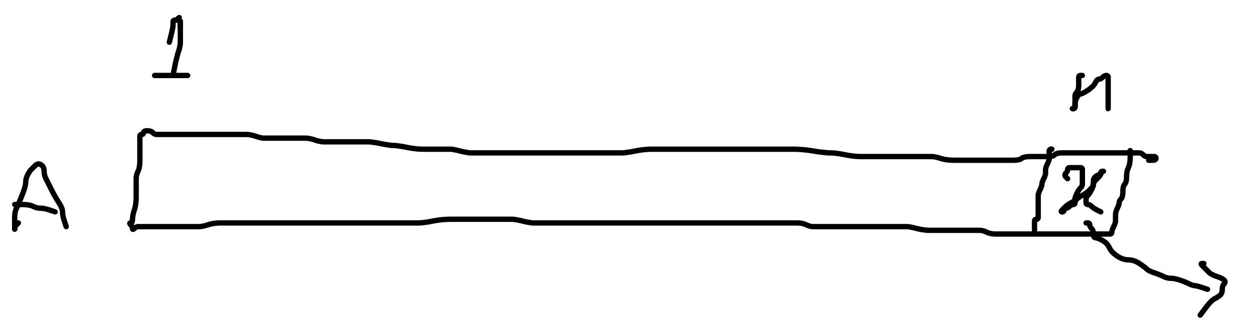
حالت متوسط : $\Theta(n \lg n)$

حل مساله مرتب سازی با روش تقسیم و غلبه

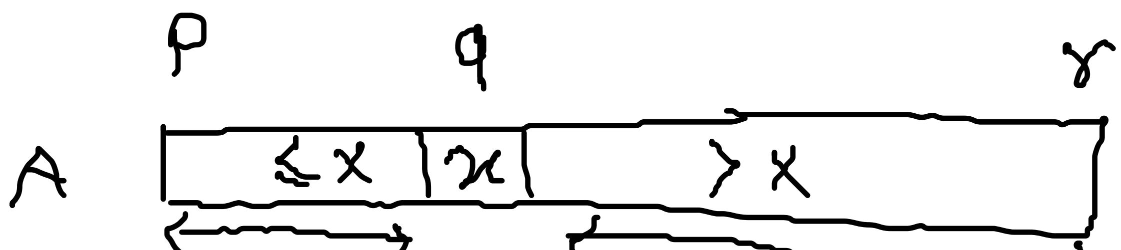
در Insertion sort بازگشتی : 1 و $n-1$ سایز زیر مساله

در merge sort : $\frac{n}{2}$ و $\frac{n}{2}$ 〃

عوض کردن روش تقسیم :

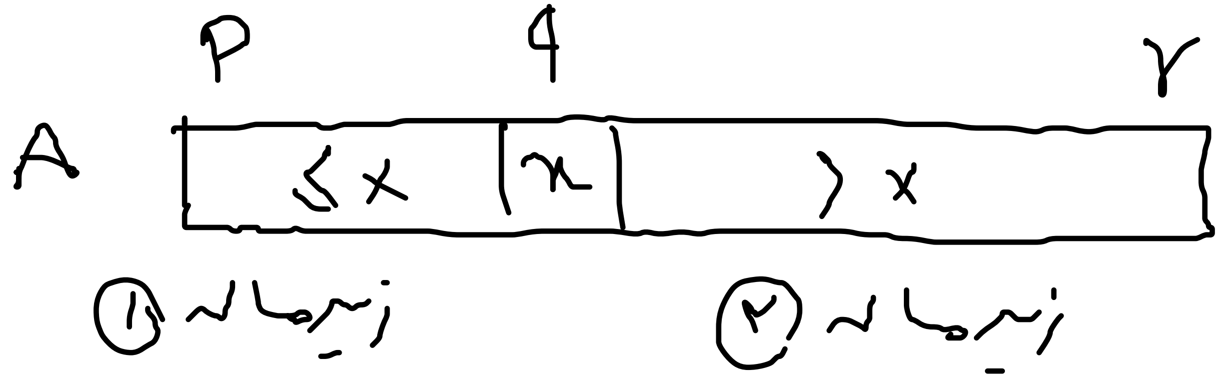


شاخص
pivot



زیرمساله 1

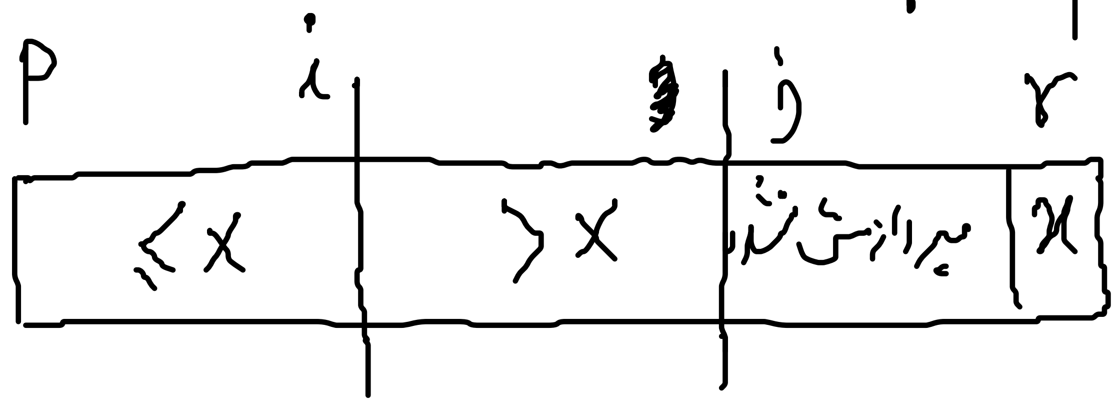
به این روش تقسیم partitioning گفته میشود



Quick_sort (A, p, r)

1. if $p \geq r$ then return
2. $q \leftarrow$ partition(A, p, r)
3. Quick_sort (A, p, q−1)
4. Quick_sort (A, q+1, r)
5. اینجا گام ترکیب نیاز نیست

نحوه انجام partition :



در هر مرحله $A[j]$ به یکی از دو بخش $>x$ یا $\leq x$ اضافه میشود:

اگر $A[j] > x \implies j++$

اگر $A[j] \leq x \implies \begin{cases} swap(A[i+1], A[j]) \\ i++ \\ j++ \end{cases}$

partition (A , p , r)

$$c'_2 < c_2$$

c_1 {
1. $x \leftarrow A[r]$
2. $i \leftarrow p-1$

$(n-1)C_2$ حد بالا (بدترین حالت) {
3. for $j \leftarrow p$ to $r-1$ do
4. if $A[j] \le x$ then
5. $i \leftarrow i+1$
6. swap ($A[i]$, $A[j]$)

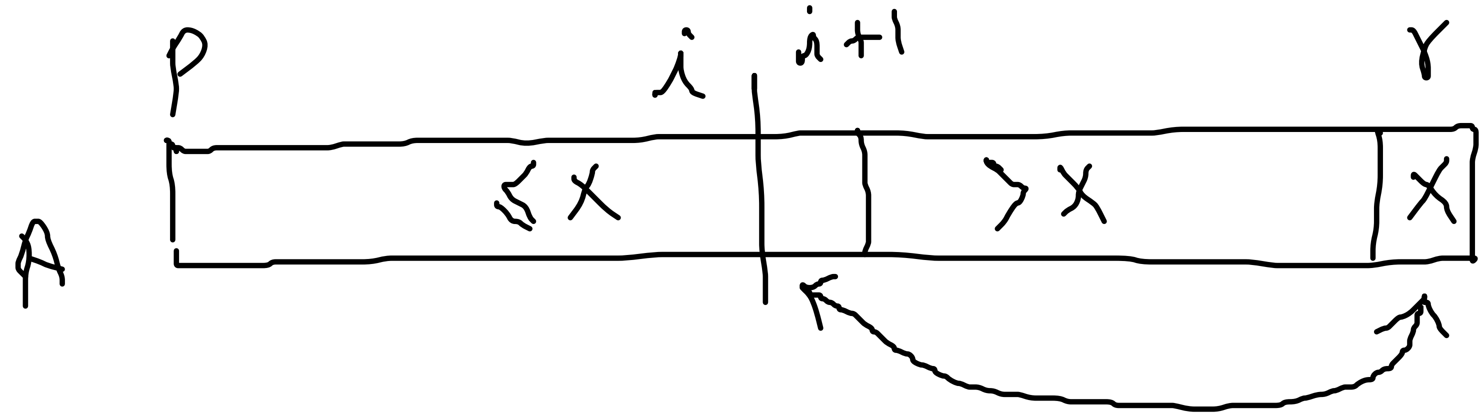
} $(n-1)\,c'_2$ حد پایین (بهترین حالت)

c_3 {
7. swap ($A[r]$, $A[i+1]$)
8. return $i+1$

partition زمان $T(n) =$

بهترین حالت: $c_1 + c'_2(n-1) + c_3 \in \theta(n)$

بدترین حالت: $c_1 + c_2(n-1) + c_3 \in \theta(n)$



partition تابع $= T(n) \in \Omega(n)$ بهترین

$\in O(n)$ بدترین حالت

$\in \Theta(n)$ حالت متوسط

زمان Quick-sort
$T(n)$

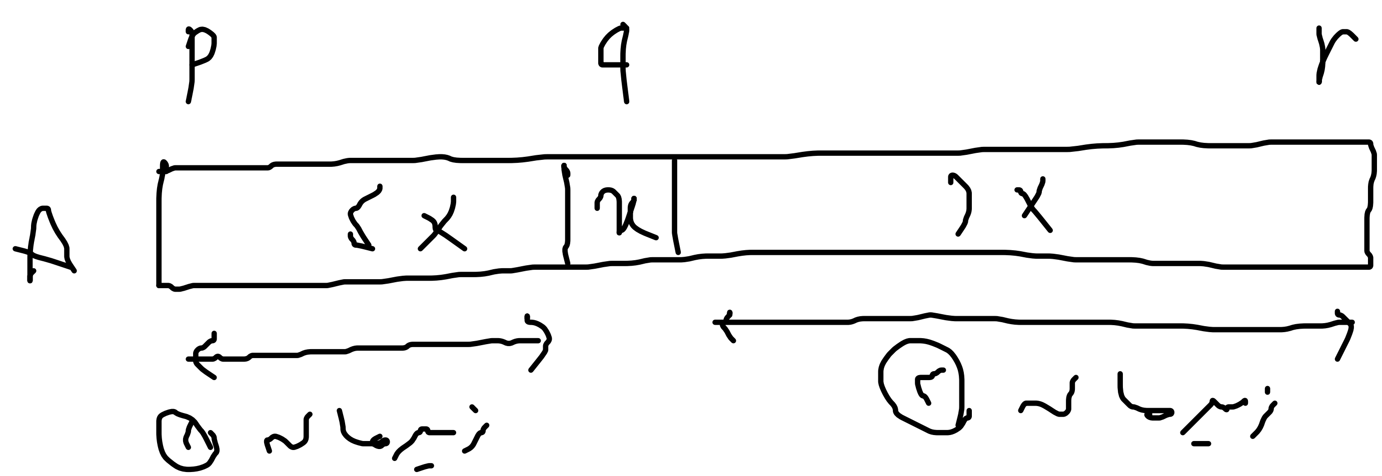
Quick-sort(A, p, r)

c_1 1. if $p \geq r$ then return

$\Theta(n)$ 2. $q \leftarrow$ partition(A, p, r)

3. Quick-sort$(A, p, q-1)$ $\longrightarrow T(k)$

4. Quick-sort$(A, q+1, r)$ $\longrightarrow T(n-k-1)$



سایز زیرمسأله 1 = ? = تعداد اعداد $\leq x$

بستگی به اعداد داخل آرایه دارد

پارامتر

1. سایز زیرمسئله $= K$
2. سایز زیرمسئله $= n - (k+1) = n - k - 1$

$$T(n) = T(k) + T(n-k-1) + \Theta(n) + c_1$$

$$= T(k) + T(n-k-1) + \Theta(n)$$

درمورد مقدار پارامتر K بحث کنیم:

$$0 \leq K \leq n-1$$

$$T(0) = T(1) = C$$

کمترین مقدار $k=0$:

$$T(n) = T(0) + T(n-1) + \theta(n)$$

$$= C + T(n-1) + \theta(n)$$

$$T(n) = T(n-1) + \theta(n)$$

روش جایگذاری

$$= [T(n-2) + \theta(n-1)] + \theta(n)$$

$$= [T(n-3) + \theta(n-2)] + \theta(n-1) + \theta(n)$$

$$\in \theta(n^2)$$

جمع قاعده حساب

بیشترین مقدار $k = n-1$:

$$T(n) = T(n-1) + T(n-(n-1)-1) + \theta(n)$$

$$= T(n-1) + T(0) + \theta(n)$$

$$= T(n-1) + c + \theta(n)$$

$$= T(n-1) + \theta(n)$$

$$\in \theta(n^2)$$

$$0 \leqslant k \leqslant n-1$$

برای حالت $k = \frac{n}{2}$ (که تقریبا "وسط" باشه

$0 \le k \le n-1$ است) :

$$T(n) = T(k) + T(n-k-1) + \Theta(n)$$

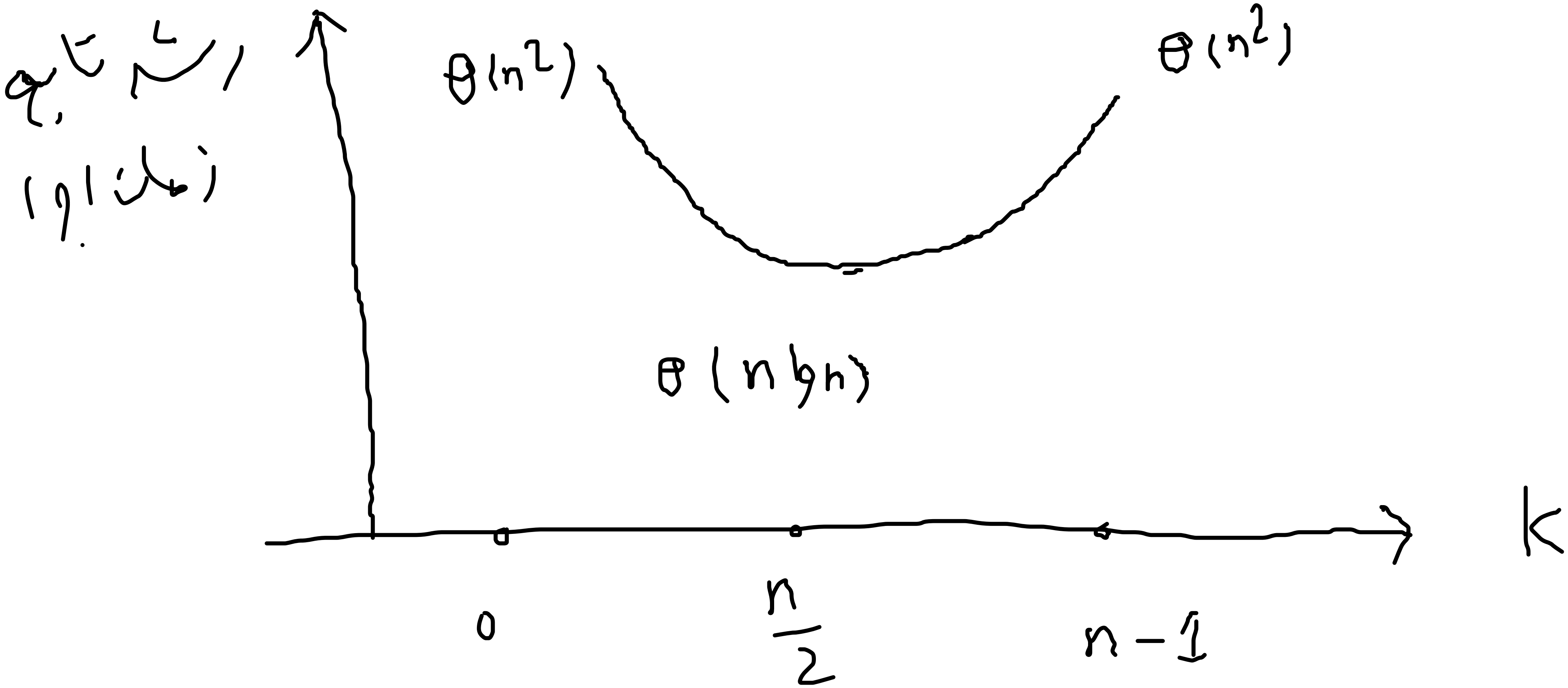
$$= T\left(\frac{n}{2}\right) + T\left(\frac{n}{2} - 1\right) + \Theta(n)$$

از نظر رشد اگر در نظر بگیریم

$$= T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + \Theta(n)$$

$$= 2T\left(\frac{n}{2}\right) + \Theta(n)$$

$$\in \Theta(n \lg n)$$

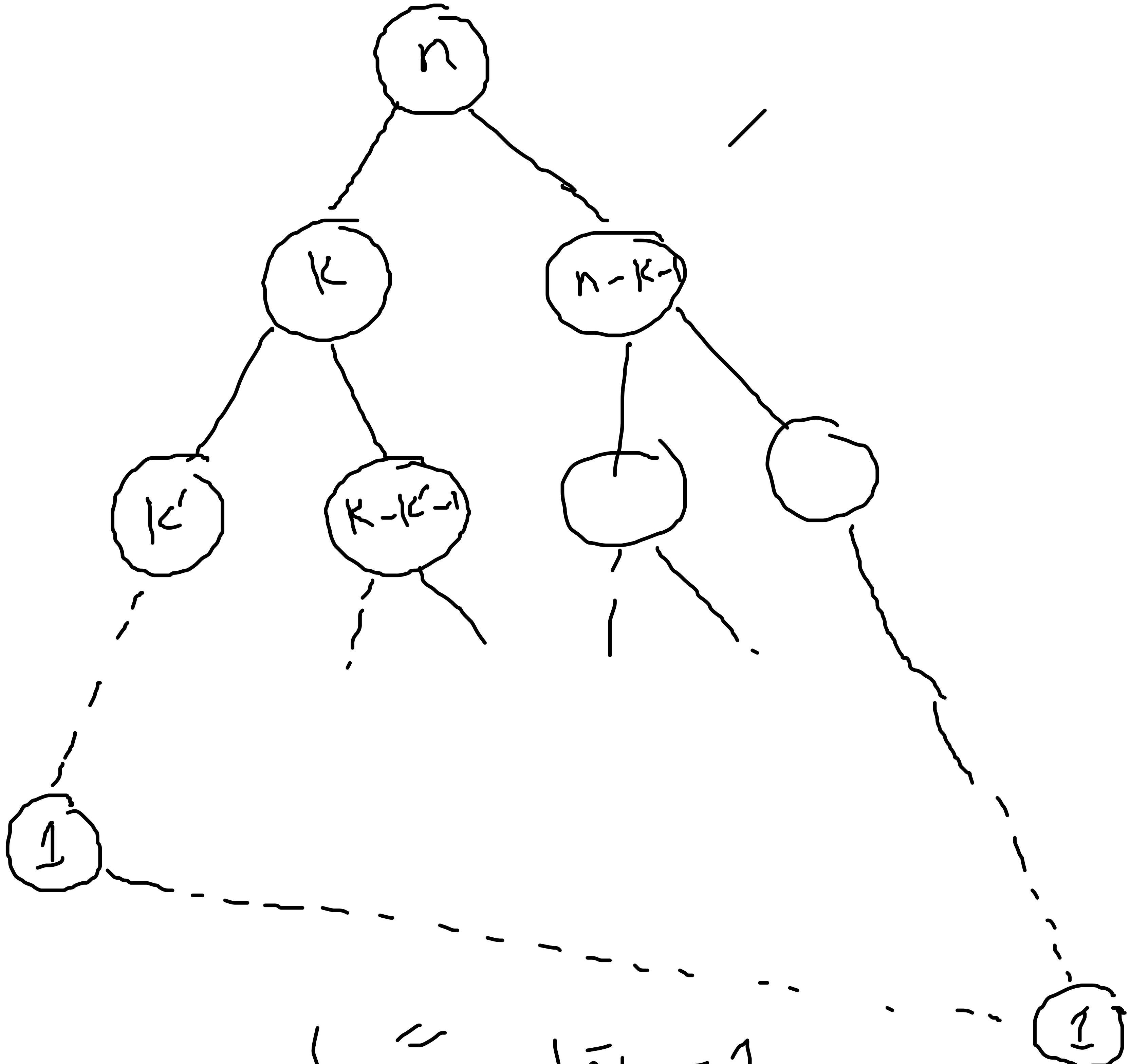


میشود نشان داد که کمترین ~~[illegible]~~ رشد تابع $T(n)$
برای وقتی رخ میدهد که اندازه ~~[illegible]~~ مساله ۲
فراخوانی ~~[illegible]~~ مساوی باشند یعنی $k \simeq \frac{n}{2}$

بنابراین می‌توان گفت که

* بهترین حالت زمان اجرا برای وقتی است که
$k = \frac{n}{2}$ باشد (یعنی سایز ۲ قسمت برابر باشد)
که در این حالت $T(n) \in \Omega(n \lg n)$

* بدترین حالت زمان اجرا برای وقتی است که
$k = 0$ یا $k = n-1$ باشد (یعنی سایز یکی از دو قسمت
صفر باشد (یا خیلی کم)) که در این حالت
$T(n) \in O(n^2)$



1 = تعداد برگ ها

حالت میانگین؟

چون سایز هر جهت (K) به این بستگی دارد

که اعداد داخل آرایه با n چه نسبتی دارند

بنابراین حالت متوسط به نحوه توزیع اعداد

در آرایه ورودی بستگی دارد.

	$1000\, n \lg n$	$2n^2$
$n=2$	2000	8
$n=4$	8000	32