۱،ۍ۵ روشی درخت بازگشت:

✓

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

$$T(n) = 4T\left(\frac{n}{2}\right) + n \quad , \quad T(1) = C$$

$$\rightarrow T\left(\frac{n}{2}\right) = 4T\left(\frac{n}{4}\right) + \frac{n}{2}$$

$$\rightarrow T\left(\frac{n}{4}\right) = 4T\left(\frac{n}{8}\right) + \frac{n}{4}$$
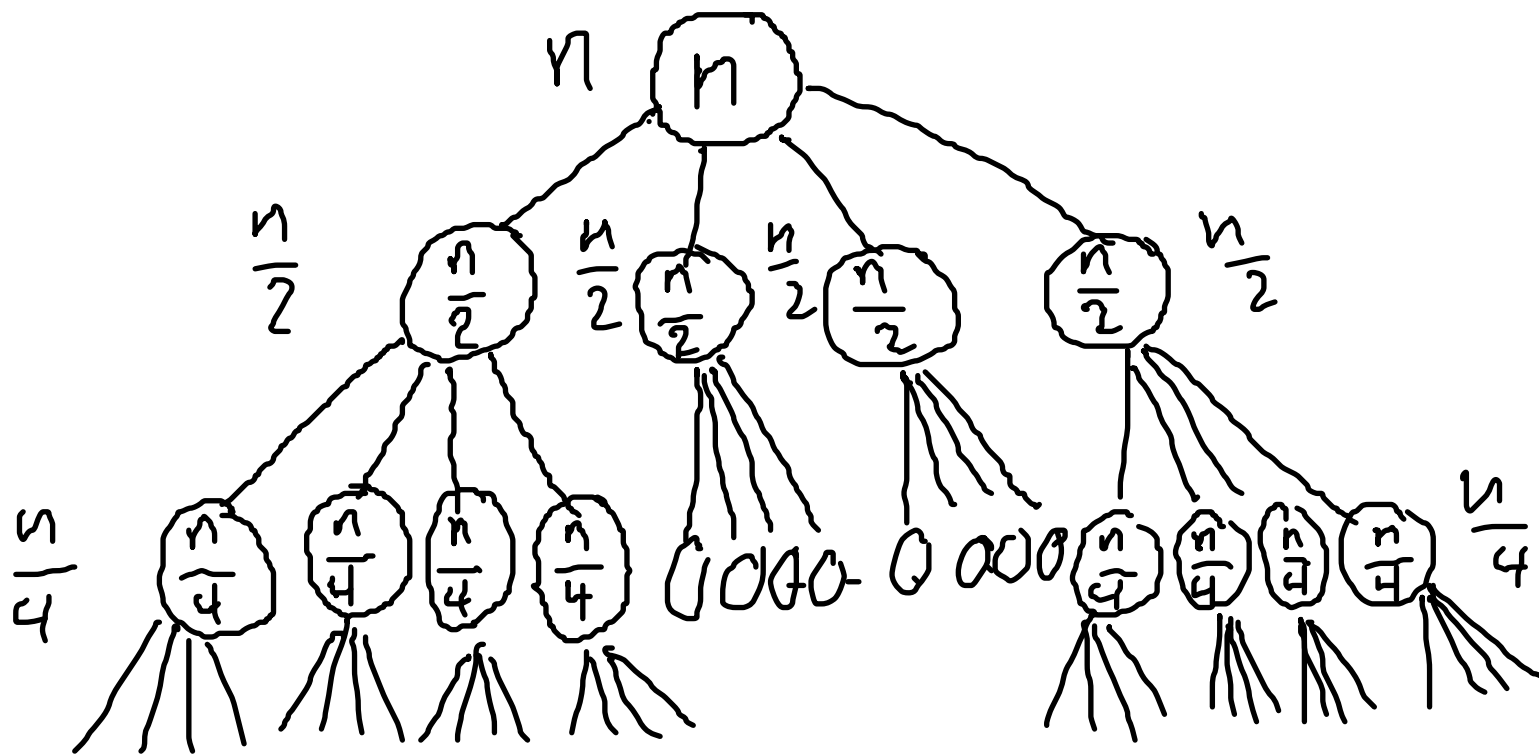
$$n$$

$$4 \times \frac{n}{2} = 2n$$

$$16 \frac{n}{4} = 4n$$

$$64 \frac{n}{8} = 8n$$

نصف ورقه کل $i = 1$

$i = 2$

$i = 3$

$i = 2$

$\frac{n}{2}$ $\frac{n}{2}$ $\frac{n}{2}$ $\frac{n}{2}$

$\frac{n}{4}$ $\frac{n}{4}$

C C

تعداد کل عمل ها

$$p \text{ در } i \text{ ام در سطح} = 2^{i-1} \times n$$

$$\text{تعداد کل عمل ها} = \sum_{i=1}^{\ell} 2^{i-1} \times n$$

نعلم بان $\dfrac{n}{2^{i-1}} = 1$ لنحسب ما

يعني بان

$$\Rightarrow \quad i-1 = \log_2 n \quad \rightarrow \quad i = \log_2 n + 1$$

لدينا اذن

$$\text{التعقيد الزمني} = \sum_{i=1}^{\log_2 n} 2^{i-1} \cdot n = n \sum_{i=0}^{\log_2 n - 1} 2^i$$

$$= n \, \frac{2^{\log_2 n} - 1}{2 - 1} = n \, \frac{2^{\log_2 n} - 1}{1} = n(n-1)$$

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

$$\text{هزینه هر مرتبه} = C \times \text{تعداد برگ‌ها}$$

$$\text{تعداد برگ در هر مرتبه} = 4^{i-1} \longrightarrow \text{تعداد برگ‌ها} = 4^{i-1} = 4^{(\log_2 \frac{n}{2}+1)-1}$$

در ریشه‌ای که $i=1$ ام

$$\text{تعداد برگ‌ها} = 4^{\log_2 \frac{n}{2}} = n^{\log_2 4} = n^2$$

$$\text{هزینه کل} = \text{هزینه برگ‌ها} + \text{هزینه داخلی}$$

$$= n^2 + n(n-1) \in \Theta(n^2)$$

روش تقسیم وحل   Divide & Conquer

① تقسیم مسأله به زیرمسأله‌های کوچکتر

۳ گام دارد ② حل بازگشتی مسائل کوچکتر

③ ترکیب جواب زیرمسائل و بدست آوردن
جواب مسأله اصلی

مساله مرتب سازی : یک آرایه از n عدد دارم

می خواهیم اعداد به صورت صعودی مرتب شوند


```
Sort_1 ( A, n )
1. if n ≤ 1 then return
2. Sort_1 ( A, n-1 )
3. ترتیب عدد آخر با n-1 عدد مرتب شده
```

گام ۱

شرط ؟

$$Key \leftarrow A[n]$$

$$i \leftarrow n-1$$

$$while\ (\ Key < A[i]\ \&\ i > 0\ )$$

$$A[i+1] \leftarrow A[i]$$

$$i \leftarrow i-1$$

$$A[i+1] \leftarrow Key$$

در واقع امین الگوریتم بالا بخشی از insertion sort

$$T(n) = T(n-1) + O(n) = \left[{}^{T(n-2) + O(n-1)}_{+O(n)} \right]$$
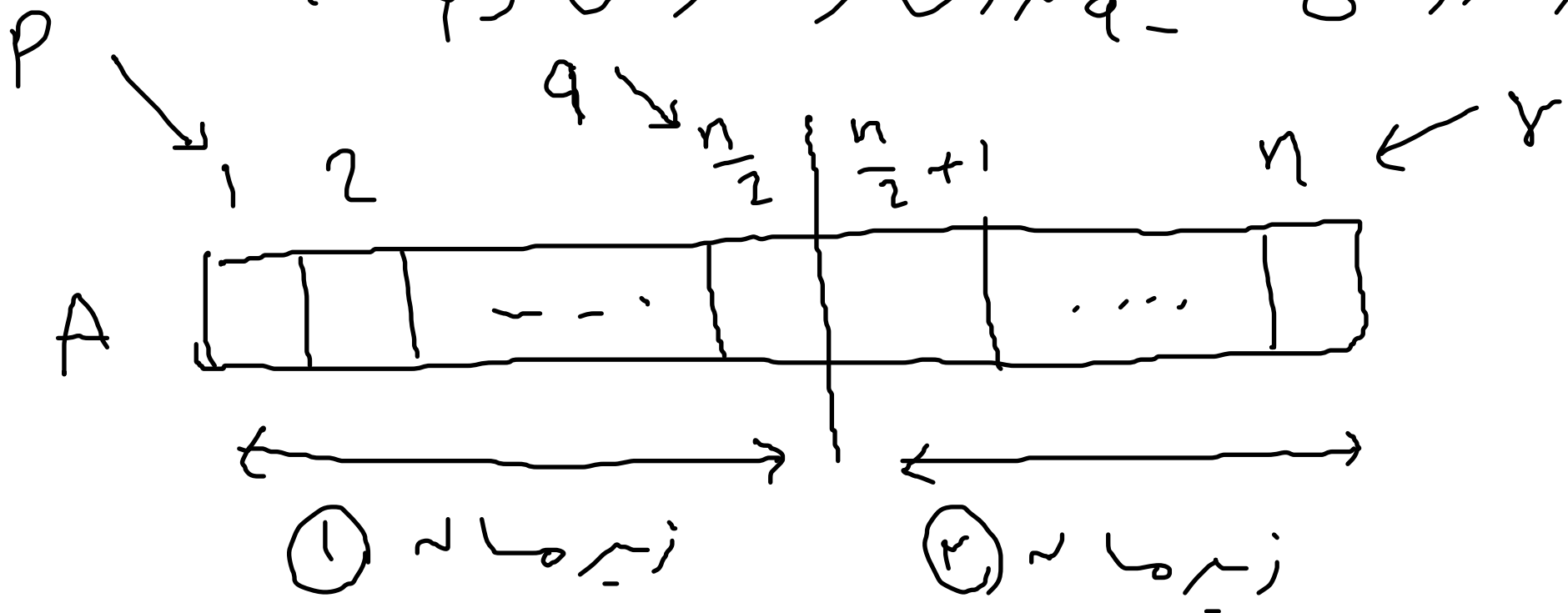
$$\in O(n^2) \qquad = c' \times \frac{n(n+1)}{2}$$

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

زبان
الگوریتم

* اگر روش تقسیم و تسخیر را عوض کنیم:

P →

q →

r ←

$$1 \quad 2 \qquad \frac{n}{2} \quad \frac{n}{2}+1 \qquad n$$

A

① زیر مسأله ② زیر مسأله

| 5 | 7 | 9 | 15 | 1 | 8 | 11 | 12 |

$$\text{Sort-2 } (A , p , r)$$

تعریف خانه وسطی

1. if $p \geqslant r$ then return

قدم ۲. $q \leftarrow \left\lfloor \dfrac{p + r}{2} \right\rfloor$

قدم بازگشتی {
3. Sort-2 (A , p , q)
4. Sort-2 $(A , q+1 , r)$

قدم ترکیب 5. merge (A , p , q , r)

این الگوریتم به جز نام هیچ فرقی با merge-sort ندارد.

i →

j →

① تعداد عناصر $n_1 = q - p + 1$

② " " $n_2 = r - q$

```
      1
      2
      3
      ⋮
  n₁
  n₁+1   ∞
      L
```

```
      1
      2
      3
      ⋮
      n₂
  ∞   n₂+1
      R
```

p ← ... → r

A

K

$\text{merge } (A, p, q, r)$

c_1 1. $n_1 \leftarrow q - p + 1, \quad n_2 \leftarrow r - q$

$c_2 n_1 \left\{ \begin{array}{l} \text{2. for } i \leftarrow 1 \text{ to } n_1 \text{ do} \\ \text{3.} \qquad L[i] \leftarrow A[p+i-1] \end{array} \right.$

c_3 4. $L[n_1 + 1] \leftarrow \infty$

$c_4 n_2 \left\{ \begin{array}{l} \text{5. for } i \leftarrow 1 \text{ to } n_2 \text{ do} \\ \text{6.} \qquad R[i] \leftarrow A[q+i] \end{array} \right.$

c_5 7. $R[n_2 + 1] \leftarrow \infty$

$\vdots$

$$c_1 + c_2 n_1 + c_3 + c_4 n_2 + c_5$$

$$= cn + c'$$

c_6 8. $i = 1$, $j = 1$

9. for $k \leftarrow p$ to r do

10.   if $L[i] < R[j]$ then
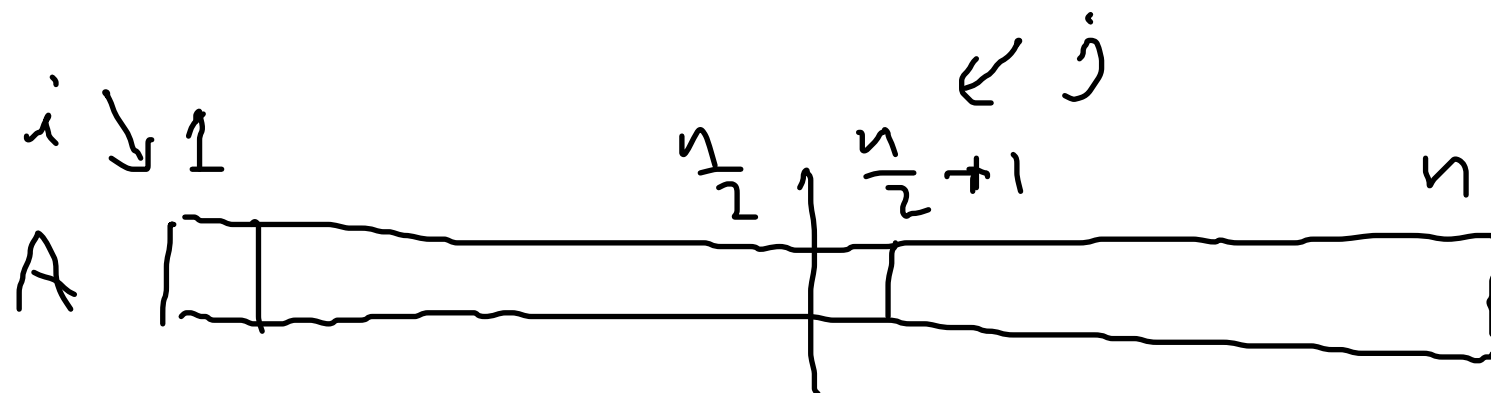
11. $A[k] \leftarrow L[i]$

12. $i++$

      else

13.

14. $A[k] \leftarrow R[j]$

15. $j++$

c_7 n spans lines 9–15

$i \searrow 1$ $\frac{n}{2}$ $\frac{n}{2}+1$ $\leftarrow j$ n

A

$$T(n) = cn + c' + c_6 + c_7 n \in \Theta(n)$$