حوری دهش ۹۸۲۱۴۱۳

مقایسه‌ها

(الف ← س۱

[۱۳, ۱, ۶, ۱۰, ۲, ۳, ۱۱, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۱

[۱, ۱۳, ۶, ۱۰, ۲, ۳, ۱۱, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۲

[۱, ۶, ۱۳, ۱۰, ۲, ۳, ۱۱, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۲

[۱, ۶, ۱۰, ۱۳, ۲, ۳, ۱۱, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۴

[۱, ۲, ۶, ۱۰, ۱۳, ۳, ۱۱, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۴

[۱, ۲, ۳, ۶, ۱۰, ۱۳, ۱۱, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۲

[۱, ۲, ۳, ۶, ۱۰, ۱۱, ۱۳, ۵, ۱۲, ۹, ۸, ۷, ۴] → ۵

[۱, ۲, ۳, ۵, ۶, ۱۰, ۱۱, ۱۳, ۱۲, ۹, ۸, ۷, ۴] → ۲

[۱, ۲, ۳, ۵, ۶, ۱۰, ۱۱, ۱۲, ۱۳, ۹, ۸, ۷, ۴] → ۵

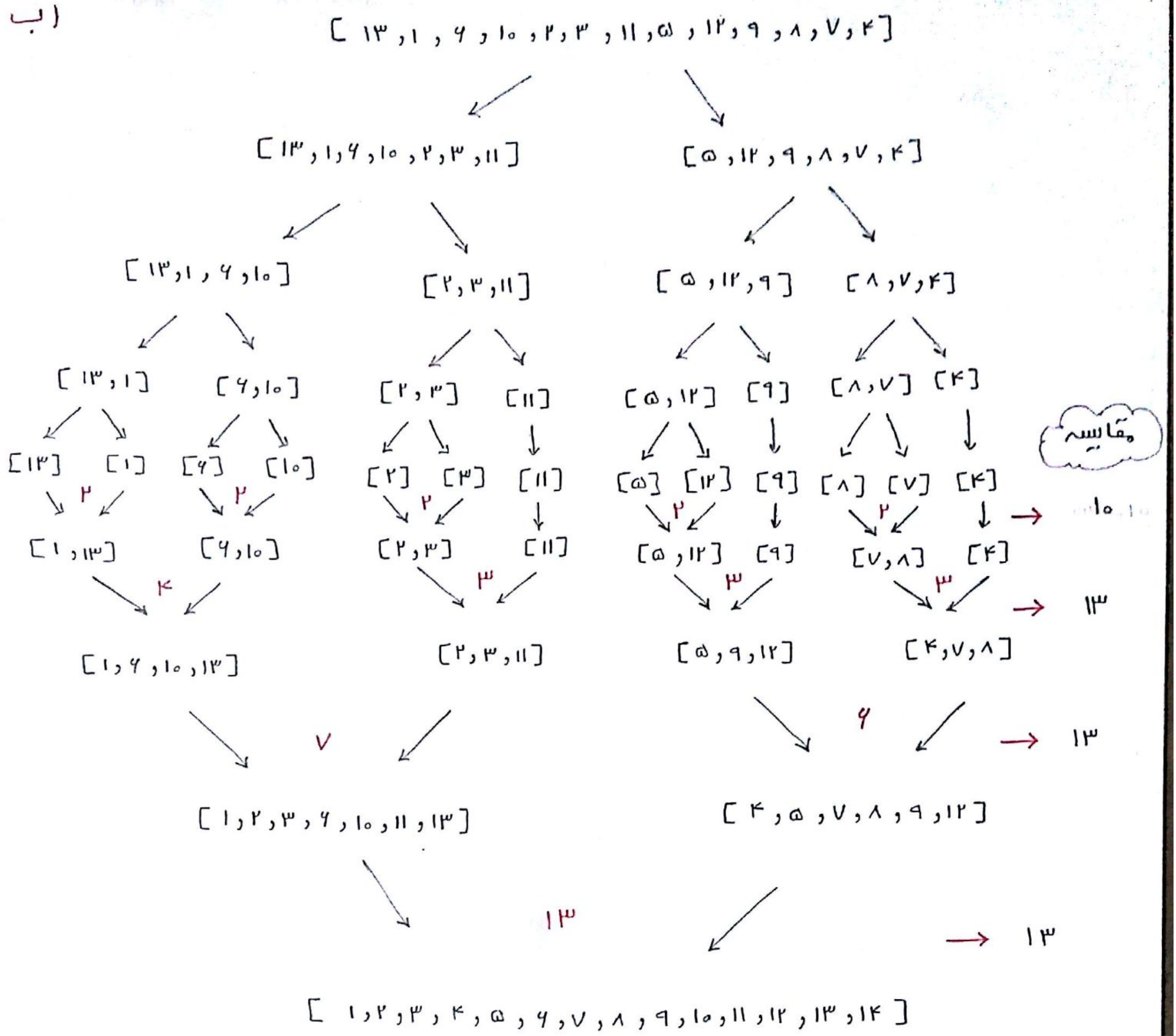
[۱, ۲, ۳, ۵, ۶, ۹, ۱۰, ۱۱, ۱۲, ۱۳, ۸, ۷, ۴] → ۶

[۱, ۲, ۳, ۵, ۶, ۸, ۹, ۱۰, ۱۱, ۱۲, ۱۳, ۷, ۴] → ۷

[۱, ۲, ۳, ۵, ۶, ۷, ۸, ۹, ۱۰, ۱۱, ۱۲, ۱۳, ۴] → ۱۰

[۱, ۲, ۳, ۴, ۵, ۶, ۷, ۸, ۹, ۱۰, ۱۱, ۱۲, ۱۳, ۱۴] → نهایی

اب



نکته ← فرض می کنیم merge sort ما آخرش بی نهایت است پس با این فرض مقایسه می کنیم .

ج)

فرض می کنیم مدت زمانی که الگوریتم برای خواباندن هر عنصر در نظر می گیرد ۱s است.

۱s: sleep ۱۳

۲s: sleep ۱

۳s: ۱ wakes up , sleep ۶

۴s: sleep ۱۰

۵s: sleep ۲

۶s: sleep ۳

۷s: ۲ wakes up , sleep ۱۱

۸s: sleep ۵

۹s: ۶ wakes up , ۳ wakes up , sleep ۱۲

۱۰s: sleep ۹

۱۱s: sleep ۸

۱۲s: sleep ۷

۱۳s: ۵ wakes up , sleep ۴

۱۴s: ۱۳ wakes up , ۱۰ wakes up

۱۷s: ۴ wakes up

۱۸s: ۱۱ wakes up

۱۹s: ۹ wakes up , ۸ wakes up , ۷ wakes up

۲۱s: ۱۲ wakes up

خروجی <= ۱ ، ۲ ، ۶ ، ۳ ، ۵ ، ۱۳ ، ۱۰ ، ۴ ، ۱۱ ، ۹ ، ۸ ، ۷ ، ۱۲

نکته = در کامپیوترهای امروزی ← سرعت خیلی بیشتر است و این به این معناست که برای خواباندن هر عنصر خیلی کمتر از ۱ ثانیه صرف می شود و ممکن است خروجی ما از این بهتر مرتب شود.

اد

پیچیدگی کل زمان sleep sort را می‌توان $O(n\log n + max(input))$ در نظر گرفت. به این صورت که:

چندین رشته بصورت داخلی توسط سیستم عامل با استفاده از صف اولویت (برای اهداف برنامه ریزی استفاده می‌شود) ایجاد می‌شود. بنابراین همهٔ عناصر آرایه در صف اولویت زمان $O(n\log n)$ را می‌گیرند.

همچنین خروجی زمانی بدست می‌آید که همهٔ نخ‌ها پردازش شوند یعنی وقتی همهٔ عناصر بیدار می‌شوند. از آنجا که بیدار شدن نخ عنصر i ام آرایه به $O(arr[i])$ زمان نیاز دارد بنابراین حداکثر $O(max\text{-}element(array))$ طول می‌کشد تا بزرگترین عنصر آرایه بیدار می‌شود.

بنابراین پیچیدگی کل زمان را می‌توان به عنوان $O(n\log n + max\text{-}element(array))$ فرض کرد که در آن ←

n = تعداد عناصر در آرایه ورودی و آرایه = عناصر آرایه ورودی

س۲ →

الف)

```
for (i=0 ; i<n ; i++) {              → n+1
      for (j=1 ; j<n ; j=j*۲) {
            cout<< i << " " ;
      }
}
```

j
1
$۲^۲$
$۲^۳$
⋮
۲^k

$۲^k \xrightarrow[\text{فرض می‌کنیم}]{} ۲^k = n \longrightarrow k = \log_۲^n$

پس $\Longrightarrow (n+1)\log^n = O(n\log^n) = T(n)$

ب)

```
int count=0;
for ( int i=n ; i>0 ; i/=۲ ) {
      for ( int j=0 ; j<i ; j++) {
            count++;
      }
}
```

داخلی‌ترین حلقه $\longrightarrow n + \frac{n}{۲} + \frac{n}{۴} + \cdots + 1 \Longrightarrow T(n) = O\left(n + \frac{n}{۲} + \cdots + 1\right) = O(n)$

ج)

```
int i,j ;
for (int i =1 ; i<=n ; i++) {
        for( j=1;  j<= log(i) ; j ++) {
                cout<< i << j ;
        }
}
```

$$T(n) = \sum_{i=1}^{n} \sum_{j=1}^{\log i} 1 = \sum_{i=1}^{n} \log i = \log ۱ + \log ۲ + \dots + \log n = \log ۲ \times ۳ \times \dots \times n = \log n!$$

د)

```
for (i=۰ ; i<n ; i++) {
        if (n %۲ ==۰) {
                for (j=i;  j>۰ ; j/=۲)     ⟶   O ( log(n-1)! )
                        cout<<"Hello";
        }
        else {
                for (k=1 ; k< i ; k*=۲)    ⟶   O ( log(n-۲)! )
                        cout<< "by";
        }
}
```

$\overset{+}{\Longrightarrow} T(n) = O(\log(n-1)!)$

برای if ⟶

i	داخلی for if
۰	۰
۱	$\log_۲ 1 = 1$
⋮	⋮
$n-1$	$\log n-1$

$\downarrow +$

$(\log(n-1)!)$

برای else ⟶

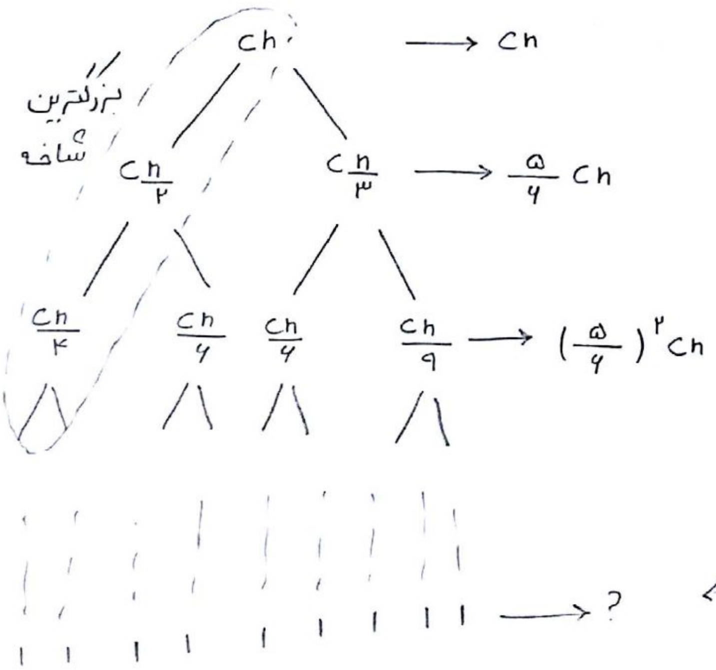
i	for
۰	۰
۱	۰
۲	$\log(۲-1)$
⋮	⋮
$n-1$	$\log(i-1)$

$+\downarrow$

$(\log(n-۲)!)$

d)
```
function a(n){
    if (n==0)
        return .....
    a(n/=2)                  → T(n/2)
    for (j=n ; j>0 ; j--)    → O(n)
        print...
    a(n/=3)                  → T(n/3)
    for (j=n; j>0 ; j--)     → O(n)
        print...
}
```

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{3}\right) + 2O(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{3}\right) + O(n)$$



ارتفاع $\rightarrow \frac{n}{2^i} = 1 \rightarrow i = \log_2^n$

تعداد برگ $\rightarrow 2^i = 2^{\log_2^n} = n^{\log_2^2} = n$

$\Rightarrow \theta(n)$

$$T(n) = cn + \frac{5}{6}cn + \cdots + \left(\frac{5}{6}\right)^{\log^{n-1}} cn + \theta(n) = \sum_{i=0}^{\log^{n-1}} \left(\frac{5}{6}\right)^i cn + \theta(n) <$$

$$\sum_{i=0}^{\infty} \left(\frac{5}{6}\right)^i cn + \theta(n) = \frac{1}{1-\frac{5}{6}} cn + \theta(n) = 6cn + \theta(n) = O(n)$$

س۳ →

a) $T(n) = ۴\,T(n/۵) + (\log n)^{۲}$

$a = ۴$ $\quad b = ۵$ $\quad f(n) = (\log n)^{۲}$

$n^{\log_b^a} = n^{\log_۵^۴}$

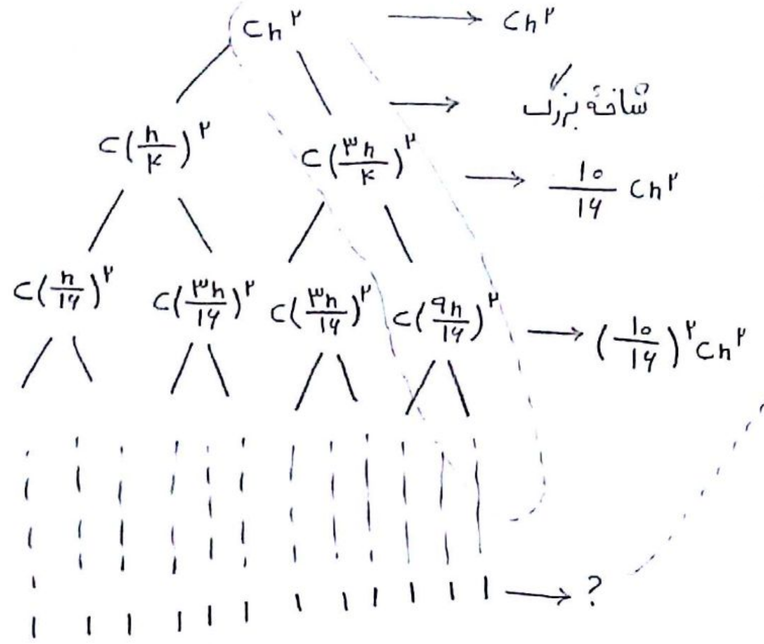
$f(n) = O\left(n^{\log_۵^۴ - \varepsilon}\right) \xrightarrow{\text{case1}} T(n) = \Theta\left(n^{\log_۵^۴}\right)$

b) $۹\,T(n) = ۹\,T(n/۳) + (\log n!)$

$a = ۹$ $\quad b = ۳$ $\quad f(n) = (\log n!)$

$n^{\log_۳^۹} = n^{۲} \longrightarrow f(n) = O\left(n^{\log_۳^۹ - \varepsilon}\right) \xrightarrow{\text{case1}} T(n) = \Theta(n^{۲})$

c) $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n^2 \quad , \quad T(1)=1$



ارتفاع $\rightarrow \dfrac{n}{\left(\frac{4}{3}\right)^i} = 1 \longrightarrow i = \log_{\frac{4}{3}}^{n}$

تعداد برگ‌ها $\rightarrow 2^i = 2^{\log_{\frac{4}{3}}^{n}} = n^{\log_{\frac{4}{3}}^{2}} \longrightarrow \Theta\left(n^{\log_{\frac{4}{3}}^{2}}\right)$

$$T(n) = cn + \frac{10}{16}cn^2 + \left(\frac{10}{16}\right)^2 + \cdots + \left(\frac{10}{16}\right)^{\log_{\frac{4}{3}}^{n-1}} + \Theta\left(n^{\log_{\frac{4}{3}}^{2}}\right) = ☆$$

$$☆ \rightarrow \sum_{i=0}^{\log_{\frac{4}{3}}^{n-1}} \left(\frac{10}{16}\right)^i cn^2 + \Theta\left(n^{\log_{\frac{4}{3}}^{2}}\right) < \sum_{i=0}^{\infty} \left(\frac{10}{16}\right)^i cn^2 + \Theta\left(n^{\log_{\frac{4}{3}}^{2}}\right) =$$

$$\frac{1}{1-\frac{10}{16}} cn^2 + \Theta\left(n^{\log_{\frac{4}{3}}^{2}}\right) = \frac{16}{6} cn^2 + \Theta\left(n^{\log_{\frac{4}{3}}^{2}}\right) = O(n^2)$$

d) $T(n) = T(\sqrt[3]{n}) + \log(n)$

$$\log n = m \longrightarrow T(2^m) = T\left(2^{\frac{m}{3}}\right) + m \xrightarrow{s(m)=T(2^m)} s(m) = s\left(\frac{m}{3}\right) + m$$

$$\xrightarrow[\text{method}]{\text{master}} \quad a=1,\ b=3,\ f(m)=m \longrightarrow m^{\log_3^1} = m^0 = 1$$

$$f(m) = \Omega\left(m^{\log_3^1 + \varepsilon}\right) \ , \ a f\left(\frac{m}{b}\right) \le c f(m) \longrightarrow \frac{m}{3} \le c\, m \ \left(c \rightarrow \frac{1}{3}\right) \xrightarrow{\text{case } 3}$$

$$s(m) = \Theta(m) \longrightarrow T(2^m) = \Theta(m) \longrightarrow T(n) = \Theta(\log n)$$

e) $T(n) = 2T(n-2) + 1 \quad , \quad T(0) = T(1) = 1$

$$T(n) = 2T(n-2)+1 = 2\left(2T(n-4)+1\right)+1 = 4\left(2T(n-6)+1\right)+2+1 =$$

$$8\left(2T(n-8)+1\right)+4+2+1 = 2^{K}\underbrace{T(n-2K)}+2^{K-1}+2^{K-2}+\dots+2+1 \Longrightarrow \star$$

اینقدر این ادامه پیدا می‌کند تا به $T(0)=1$ برسد

$$n-2K=0 \longrightarrow K=\frac{n}{2}$$

$$\star \rightarrow 2^{K}+2^{K-1}+\dots+2+1 = \sum_{i=0}^{K} 2^{i} = 2^{K+1}-1$$

$$\longrightarrow T(n) = O\left(2^{K+1}\right) \xrightarrow{K=\frac{n}{2}} O\left(2^{\frac{n}{2}+1}\right) = O(2^{n})$$

f) $T(2^{K}) = 7T(2^{K-2}) + 2^{K}$

$$2^{K} = m \longrightarrow T(m) = 7T\left(\frac{m}{4}\right)+m$$

master method $\longrightarrow$ $a=7$, $b=4$, $f(m)=m$ $\longrightarrow m^{\log_4 7}$

$$f(m) = O\left(m^{\log_4 7-\varepsilon}\right) \xrightarrow{\text{case1}} T(m) = \Theta\left(m^{\log_4 7}\right) \longrightarrow T(2^{K}) = \Theta\left((2^{K})^{\log_4 7}\right) =$$

$$\Theta\left(2^{\log_4 7^{K}}\right)$$

س۴ →

$O(f+g) = O(f) + O(g) \longrightarrow$ درست

$O(f \cdot g) = O(f) \cdot O(g) \longrightarrow$ درست

if $g = O(f)$ and $h = O(f)$ then $g = O(h) \longrightarrow$ نادرست

↳ if $g = O(f)$ and $f = O(h)$ then $g = O(h)$

$5n^2 + 100n^3 + 4n = O(n^4) \longrightarrow$ درست

$5n^2 + 100n^3 + 4n = O(n^2 \log n) \longrightarrow$ نادرست $\longrightarrow 5n^2 + 100n^3 + 4n = \Omega(n^2 \log n)$

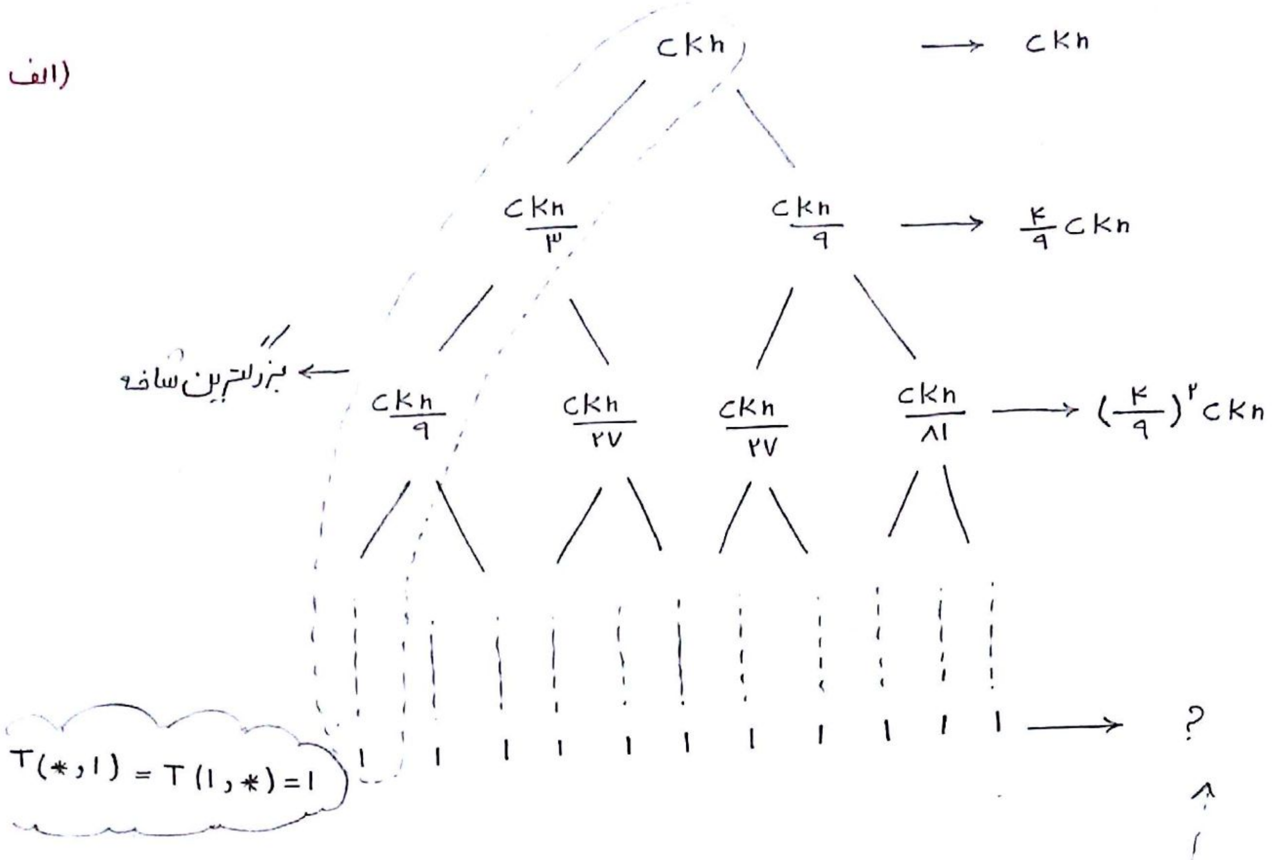
س۵ →

$n^n > n^4 \binom{n}{4} > \pi^n > n^\pi > n! > 2^n > n^2 > 2^{\log^4 n} > (\log n)^n > n \log n >$

$\sqrt{2^{\sqrt{n}}} > (\log n)^{\log n}$

س۶ →

الف)



ارتفاع → $\frac{Kn}{3^i} = 1 \rightarrow i = \log_3^{Kn}$

ب)

تعداد برگ‌ها → $2^i = 2^{\log_3^{Kn}} = Kn^{\log_3^2} \rightarrow \Theta(Kn^{\log_3^2})$

$$T(n,K) = cKn + \frac{4}{9}cKn + \left(\frac{4}{9}\right)^2 cKn + \dots + \left(\frac{4}{9}\right)^{\log_3^{Kn-1}} + \Theta(Kn^{\log_3^2}) =$$

$$\sum_{i=0}^{\log_3^{Kn-1}} \left(\frac{4}{9}\right)^i cKn + \Theta(Kn^{\log_3^2}) < \sum_{i=0}^{\infty} \left(\frac{4}{9}\right)^i cKn + \Theta(Kn^{\log_3^2}) = \frac{1}{1-\frac{4}{9}} cKn +$$

$$\Theta(Kn^{\log_3^2}) = \frac{9}{5}cKn + \Theta(Kn^{\log_3^2}) = O(Kn)$$

س۸ →

الف)

$$H_n = \sum_{i=1}^{n} 1/i = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

$$H_n \geq 1 + \frac{1}{2} + \underbrace{\frac{1}{4} + \frac{1}{4}}_{\frac{1}{2}} + \underbrace{\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}}_{\frac{1}{2}} + \cdots = 1 + \frac{1}{2} \lceil \log n \rceil$$

$$\Longrightarrow \star$$

$$H_n \leq 1 + \underbrace{\frac{1}{2} + \frac{1}{2}}_{1} + \underbrace{\frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}}_{1} + \frac{1}{8} + \cdots = 1 + \lfloor \log n \rfloor$$

$$\star \rightarrow H_n = \theta(\log n)$$