سوال 1
هنوز پاسخ داده نشده است
نمره از 5.00

الگوریتم مرتب سازی زیر را روی آرایه A با n عضو که مقادیر آنها به صورت یکتا از 1 تا n است اجرا می کنیم. صحت عبارات زیر را تعیین کنید.

```
INDEXSORT(A)
    for i ← 1 to n
        do while A[i] ≠ i
            do SWAP (A[i], A[A[i]])
```

تعداد جابجایی ها همیشه برابر با تعداد عناصری است که سرجای خودشان نیستند.	غلط
آرایه ای وجود دارد که الگوریتم دقیقا n-1 جابجایی انجام دهد.	صحیح
الگوریتم به ازای هر ورودی n-1 جابجایی انجام خواهد داد.	غلط
بیشترین تعداد جابجایی ها زمانی است که آرایه به صورت برعکس مرتب شده باشد.	غلط
این الگوریتم مرتب سازی را بدرستی انجام خواهد داد.	صحیح

1 n n-1

۱ ۴ ۲ ۳ ۵

۱ ۳ ۲ ۴ ۵

۱ ۲ ۳ ۴ ۵

۵ ۴ ۳ ۲ ۱ ۱ ۲ ۳ ۴ ۵

A[۵]

زمان اجرای کد زیر چیست؟

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

a. O(lg n)

b. O((lg n)^2)

c. O(n^2)

d. O(n)

e. O(n lg n)

N

$$\overbrace{N \quad N-1 \quad N-2 \quad N-3 \quad \dots \quad 1}^{n}$$

$$\frac{n(n+1)}{2}$$

زمان اجرای کد زیر چیست؟

```
int a = 0;
for (i = 0; i < N; i++) {      → n
    for (j = N; j > i; j--) {  → n - i
        a = a + i + j;
    }
}
```

- ○ .a O((lg n)^2)
- ● .b O(n^2)
- ○ .c O(lg n)
- ○ .d O(n lg n)
- ○ .e O(n)

n/۲ lg n

زمان اجرای کد زیر چیست؟

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

lg n

- a. O(lg n)
- b. O(n^2)
- c. O((lg n)^2)
- d. O(n lg n)
- e. O(n)

$a(\frac{T}{b}) + n^k \lg^p n$

$\lg_{۴}^{۳}$ k

a(t/b)+n^k log ^p n a/b and k and p>1

جواب رابطه بازگشتي زير چيست؟

$$T(n) = 3T(n/4) + n \log n$$

$T(n) = \Theta(n^k \lg^p n)$

- a. Θ(n^2 lg n)
- b. Θ(n^3)
- c. Θ(n^2)
- d. Θ(n)
- e. Θ(n lg n)

a = ۳ k = ۱ $\lg_{۴}^{۳}$ k = ۱

b = ۴ p = ۱

$T(n) = aT(n/b) + \theta(n^k \log^p n)$

۸۱ = a

۹ = b

$a = b$^۲

$f(n) = n$^۲

$\log_b a = \Theta f(n)$

$\Rightarrow n^{\log_b a} \lg n$

با استفاده از قضیه اصلی، بهترین و دقیق ترین زمان اجرای رابطه زیر چه خواهد بود؟

T(n)=81T(n/9)+O(n^2)

- a. O((n^2)lgn)
- b. O(n^2)
- c. O(nlgn)
- d. O(n^4)
- e. O(n)
- f. O((n^3)lgn)
- g. O(n^3)
- h. O((n^4)lgn)

n^k

$\lg^{۲۷}_{/۹}$

با استفاده از قضیه اصلی، بهترین و دقیق ترین زمان اجرای رابطه زیر چه خواهد بود؟

T(n)=27T(n/9)+O(n^2)

- a. O((n^4)lgn)
- b. O(n^4)
- c. O(nlgn)
- d. O(n^2) ✓
- e. O(n^3)
- f. O((n^2)lgn)
- g. O(n)
- h. O((n^3)lgn)

۲

با استفاده از قضیه اصلی، بهترین و دقیق ترین زمان اجرای رابطه زیر چه خواهد بود؟

$T(n)=25T(n/5)+O(n)$

- a. $O(n^2)$
- b. $O((n^3)lgn)$
- c. $O(n^3)$
- d. $O(n)$
- e. $O((n^2)lgn)$
- f. $O(nlgn)$
- g. $O(n^4)$
- h. $O((n^4)lgn)$

اگر دو تابع f(x) و g(x) به صورت زیر تعریف شده باشند، آنگاه بهترین رابطه ای که برای نمایش نسبت آنها به هم می توان نوشت، کدام گزینه است؟

f(x)=4n^3+12n^2+42

g(x)=32n^2+52n^3+42

$$\lim \frac{g(n)}{f(n)} = \frac{۵۲}{۴} = c$$

۴

- a. f=Θ(g)
- b. f=O(g)
- c. f=Ω(g)

$$\frac{۵۲n}{۴} = cn$$

۳

الگوریتم زیر چه کاری انجام می دهد و پیچیدگی زمانی آن چیست؟

```
SS(A[0 . . n-1])
   If  n = 2 and A[0] > A[1] then
      Swap (A[0] , A[1])
else if n > 2
      m= ⌈2n/3⌉
      SS(A[0. . m-1])
      SS(A[n-m. .n-1])
      SS(A[0. . m-1])
```

۳ (۲n/۳) + O(۱)

- a. آرایهٔ A را لزوماً مرتب نمی‌کند اما زمان اجرای آن $\theta\left(n^{\log_{۲/۳} ۳}\right)$ است.
- b. آرایهٔ A را مرتب می‌کند و زمان اجرای آن $\theta\left(n^{\log_{۳/۲} ۳}\right)$ است.
- c. آرایهٔ A را لزوماً مرتب نمی‌کند اما زمان اجرای آن $\theta\left(n^{\log_{۳/۲} ۳}\right)$ است.
- d. آرایهٔ A را مرتب می‌کند و زمان اجرای آن $\theta\left(n^{\log_{۲/۳} ۳}\right)$ است.

۳

$$f(n) = f(n-1) + g(n)$$
$$g(n) = f(n-1) + g(n/2)$$

$$f(n) = 2f(n-1) + g(n/2)$$

$$f(n) > 2f(n-1)$$

$$f(n) = 2 \times 2 \times (f(n-2) + g(n/2) + \dots$$

$$f(n) = \underline{2^k} f(n-k) + \dots$$

... کدام دسته می تواند باشد؟

```
int f(i
{
    if
    els
}

int g(i
{
    if
    els
}
```

Question 111 Explanation:

f(n) = f(n-1) + g(n) ---- 1

g(n) = f(n-1) + g(n/2) ---- 2

Putting the value of g(n) in equation 1,

f(n) = 2*f(n-1) + g(n) + g(n/2)

So, we can derive the below equation,

f(n) > 2f(n-1)

=> f(n) > 2*2*f(n-2) ---- because f(n) > 2*f(n-1), so, f(n-1) > 2*f(n-2)....

So we can write f(n)>2*2*f(n-2).

so on....

=> f(n) > (2^n)f(1) --- here '^' denotes the exponent.

So, f(n) > Theta(2^n)

So, option B is true, exponential growth for f(x).

- a. درجه 2
- b. لگاریتمی
- c. خطی
- d. ثابت
- e. نمایی
- f. درجه 3

اگر در الگوریتم مرتب سازی ادغامی، در هر مرحله به جای تقسیم آرایه به دو بخش مساوی، آن را به دو قسمت به صورت تصادفی تقسیم کنیم. میانگین زمان اجرای این الگوریتم چه تغییری می کند؟ و در بدترین حالت چه پیچیدگی خواهد داشت؟

- a. میانگین زمان از الگوریتم اولیه سریعتر خواهد بود و در بدترین حالت نیز از O(n^2) خواهد بود.
- b. میانگین زمان از الگوریتم اولیه سریعتر خواهد بود و در بدترین حالت نیز از O(nlgn) خواهد بود.
- c. میانگین زمان از الگوریتم اولیه کندتر خواهد بود و در بدترین حالت نیز از O(n^2) خواهد بود.
- d. میانگین زمان تغییری نخواهد کرد و در بدترین حالت نیز از O(n^2) خواهد بود.
- e. میانگین زمان از O(n) خواهد بود و در بدترین حالت نیز از O(n^2) خواهد بود.
- f. میانگین زمان از الگوریتم اولیه کندتر خواهد بود و در بدترین حالت نیز از O(nlgn) خواهد بود.
- g. میانگین زمان تغییری نخواهد کرد و در بدترین حالت نیز از O(nlgn) خواهد بود.

کدام مورد برای رابطه بازگشتی زیر صحیح است؟

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{6}\right) + n^{\sqrt{\lg n}}$$

به ازای مقادیر کوچک n مقدار $T(n) = 1$ است.

a. $\theta\left(n^{\sqrt{\lg n}} \lg n\right)$

b. $\theta\left(n^{\sqrt{n}}\right)$

c. $\theta\left(n^{\sqrt{n \lg n}}\right)$

d. $\theta\left(n^{\sqrt{n}} \lg n\right)$

e. $\theta\left(n^{\sqrt{\lg n}}\right)$

n

$\frac{n}{3}$ $\frac{n}{6}$

کدام مورد برای رابطه بازگشتی زیر صحیح است؟

$$T(n) = 2^n T(n-1)\,, T(0) = 1$$

- ○ a. $\theta\left(2^{\sqrt{n\,lg\,n}}\right)$
- ○ b. $\theta\left(\sqrt{2}^{n^2}\right)$
- ○ c. $\theta\left(\sqrt{2}^{n^2+n}\lg n\right)$
- ○ d. $\theta(n\,2^n)$
- ◉ e. $\theta\left(\sqrt{2}^{n^2+n}\right)$ ✓
- ○ f. $\theta(2^n \lg n)$

$T(n) = ۲^n . T(n-۱)$

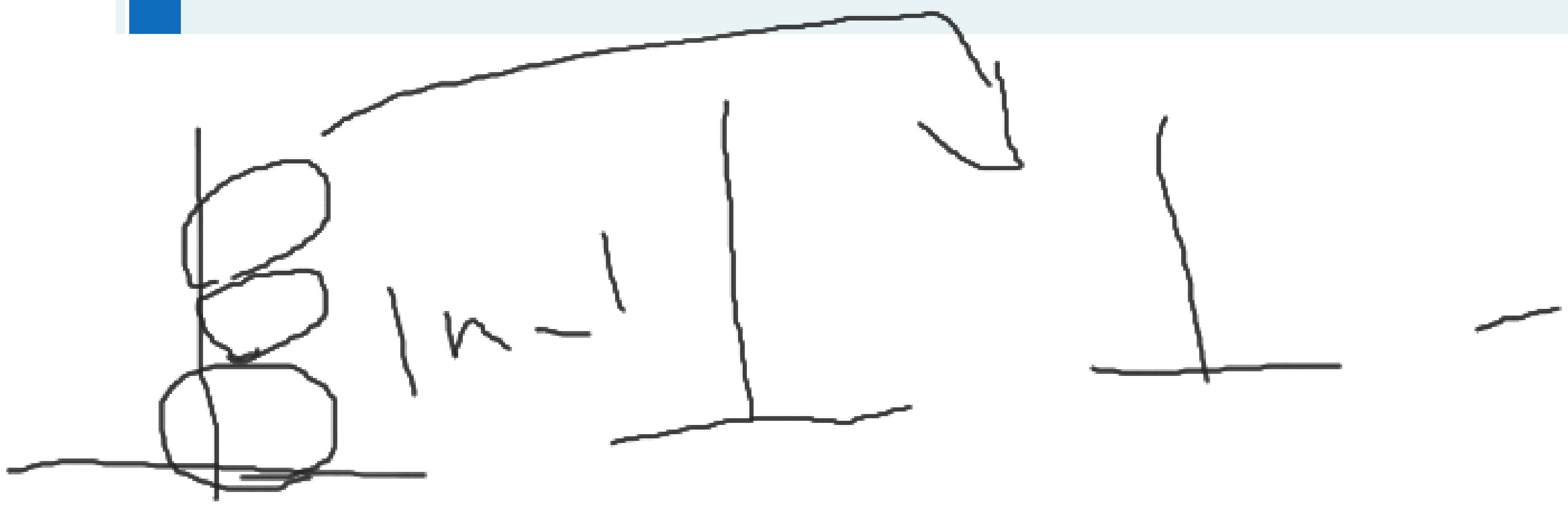
$T(n) = ۲^n\, ۲^{n-۱}\, T(n-۲)$

$۲^n \times ۲^{n-۱} \times \ldots ۲^۰$

برای پیدا کردن کمترین تعداد جابجایی لازم دیسکها در مساله برج های هانوی به صورت بازگشتی، کدام رابطه بازگشتی می تواند مفید باشد؟

(مساله برج های هانوی دارای 3 میله A,B,C است که تعداد n دیسک روی میله A به صورت بزرگ به کوچک از پائین به بالا قرار گرفته اند. برای حل مساله باید تمامی دیسکها از A به B منتقل شوند اما دو شرط داریم. در هر جابجایی فقط یک دیسک قابل جابجا شدن است. هیچ وقت نباید دیسک سنگین روی دیسک سبک قرار گیرد.

- ○ .a $T(n)=T(n-1)+T(n-2)+3$
- ◉ .b $T(n)=2T(n-1)+1$
- ○ .c $T(n)=T(n-1)+T(n-2)+1$
- ○ .d $T(n)=6T(n-1)+3$
- ○ .e $T(n)=T(n-1)+T(n-2)+2$



اگر a=n^(2*cosα) و b=n^3، آنگاه کدام رابطه زیر درست است؟

- a. $a=\Theta(b)$
- b. $a=O(b)$
- c. $a=\Omega(b)$

جواب رابطه بازگشتی زیر چیست؟

$$T(n) = 3T(n/3) + \sqrt{n}$$

- a. $\Theta(n^3)$
- b. $\Theta(n^2 \lg n)$
- c. $\Theta(n \lg n)$
- d. $\Theta(n)$
- e. $\Theta(n^2)$

۳
۱ ۹ ۲

۱/۲

در مسئله ضرب دو عدد n رقمی به صورت تقسیم و غلبه که در کلاس مطرح شد، رابطه بازگشتی زمان اجرا به چه صورت است؟

- a. $T(n)=4T(n/2)+O(n)$
- b. $T(n)=T(n-1)+O(n)$
- c. $T(n)=3T(n/2)+O(n)$
- d. $T(n)=2T(n/2)+O(n)$
- e. $T(n)=3T(n/3)+O(n)$

ya bedonid o n^2 e va bezanid ya tahlil konid

https://www.nayuki.io/page/master-theorem-solver-javascript