Subject:
Date

1)

الف) int i,j,k = 0

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

حلقه درونی $\log_2 n$ بار اجرا می‌شود

حلقه بیرونی $\frac{n}{2}$ بار اجرا می‌شود

$n \to \infty$

$$O\left(\frac{n}{2}\log_2 n\right) = \boxed{\log\left(n \log_2 n\right)}$$

Subject: ۲
Date

11 7 5
27.29 18.6

```
int i=n
while(i>1) {
    int j=1
    while(j<n) {
        j=j*5
        i= i/3
        cout << "*"
    }
}
```

$\log_3^n = 5$

$\log_5^n = 3$

$\frac{i}{(3)^x} < 1$

$\boxed{i < (3)^n}$

$\log_5^n \, x$

حلقه درونی هر بار $\log_5^n$ بار تکرار می‌شود

حلقه بیرونی به علت تداخل با حلقه داخلی باید بررسی بیشتری قرار می‌گیرد

$i=i/3$ می‌دانیم اگر حلقه داخلی فقط یکبار اجرا می‌شد

آن وقت حلقه بیرونی $\log_3^n$ بار اجرا می‌شود

پس می‌توان گفت در کل $\lceil \log_3^5 \rceil \times (\log_5^n)$ بار برنامه اجرا می‌شود

پس $O(\log_5^n)$

$O(2\log_5^n) = O(\log_5^n)$

Subject: ادامه سوال 1
Date

2)
```
Void function(int n){
    int count = 0;
    for(int i=0; i<n; i++)          ①
        for(int j=1; j<i*i; j++)    ②
            if(j%i==0){
                for(int k=0; k<j; k++)   ③
                    print("*");
            }
}
```

حلقه ① n بار اجرا می‌کند

حلقه ② به مقدار $1^2+2^2+3^2 \dots n^2$ اجرا می‌کند

حلقه ③ (اگر به صورت ~~...~~ worst case در نظر بگیریم همواره اجرا شده داریم:

$$T = \sum_{i=0}^{n}\sum_{j=1}^{i^2}\sum_{k=0}^{j} 1 = \sum_{i=0}^{n}\sum_{j=1}^{i^2}(1+j)$$

ادامه در صفحه بعد

Subject:

Year: Month: Date:

$$T = \sum_{i=0}^{n-1} \sum_{j=1}^{i^2-1} \sum_{k=0}^{j-1} 1 = \sum_{i=0}^{n-1} \sum_{j=1}^{i^2-1} j$$

$$= \sum_{i=0}^{n-1} i^3 - \frac{i^2}{2} - \frac{i}{2}$$

$$\simeq \left(\frac{1}{4} n^2 (n-1)^2 - \frac{1}{6} (n-1) n \right) (2n+1) \simeq O(n^4)$$

$$O(n^4)$$

Subject:
Date

2)

الف) $\log f(n) \in \Theta(\log g(n)) \Rightarrow f(n) \in \Theta(g(n))$ ✗

مثال نقض

$f(n) = n^4$
$g(n) = n^2$ $\Rightarrow \log n^4 \in \Theta(\log n^2)$

$4 \log n \in \Theta(2 \log n)$

$\log n \in \Theta(\log n)$

اما:

$n^4 \notin \Theta(n^2)$

رد می‌شود

ب) $f(n) \in \Omega(g(n)),\ g(n) \in \Omega(h(n)) \Rightarrow f(n) \in \Omega(h(n))$

درست ✓

$f(n) \in \Omega(g(n)) \Rightarrow f(n) \geq c\,g(n)$

اثبات

طبق تعریف داریم

$\Omega(g(n)) = \{ f(n) : \exists c, n_0 > 0$ such that $\forall n \geq n_0,\ 0 \leq c\,g(n) \leq f(n) \}$

حال می‌دانیم

$f(n) \in \Omega(g(n)) \Rightarrow \exists c_1, n_1 > 0$ such that

$\forall n > n_1$ ~~$0 \leq f(n) \leq$~~ $0 < c_1 g(n) \leq f(n)$

$g(n) \in \Omega(h(n)) \Rightarrow \exists c_2, n_2 > 0$ such that

$\forall n > n_2$ $0 < c_2 h(n) \leq g(n)$

Subject:
Date

ادامه سوال 2 بخش آ

حال اگر $\boxed{n_3 = \max(n_1, n_2)}$ و $\boxed{c_3 = c_1 \times c_2}$ داریم

$$\forall n > n_3 \quad 0 \le c_1 c_2 h(n) \le c_1 g(n) \le f(n)$$

$$\Rightarrow \forall n > n_3 \quad 0 \le c_3 h(n) \le c_1 g(n) \le f(n)$$

$$\Rightarrow \forall n > n_3 \quad 0 \le \underbrace{c_3 h(n) \le f(n)}$$

$$\Rightarrow f(n) \in \Omega(h(n)) \checkmark$$

ج:

$f(n) \in O(g(n))$ or $f(n) \in \Omega(g(n))$ or $f(n) \in \Theta(g(n))$ which one ?

می‌توان گفت برای یک n و c به بعد تابع $f(n)$ و $g(n)$ می‌تواند به سه حالت باشد یعنی

1. $a \le b \Leftarrow f(n) \le g(n)$
2. $a = b \Leftarrow f(n) = g(n)$
3. $a \ge b \Leftarrow f(n) \ge g(n)$

خوب می‌دانیم که همواره به ازای $n \to \infty$ و c بایستی یکی از احتمالات $f(n)$, $g(n)$ برقرار شود و یکی از این سه حالت بالا رخ دهد (چون امکان ندارد که متفاوتی داشته باشد) پس می‌توان نوشت

$f(n) \in O(g(n))$

or

$f(n) \in \Omega(g(n))$

or

$f(n) \in \Theta(g(n))$

Subject:

Year: Month: Date:

الف) ج)

$$T(n) = T(\sqrt{n}) + O(\underbrace{\log \log n}_{m})$$

تغییر متغیر

$$T(2^{2^m}) = T(2^{2^{m-1}}) + O(m)$$

$$T(2^{2^m}) = f(m)$$

$$\Rightarrow f(m) = f(m-1) + O(m) = f(m-2) + O(m-1) + O(m)$$

$$\hookrightarrow f(m) = O(m^2)$$

$$\hookrightarrow T(2^{2^m}) = O(m^2) \Rightarrow \boxed{T(n) = O((\log(\log n))^2)}$$

Subject:
Date

حل ادامه

ب) $T(n) = 25\,T\left(\frac{n}{5}\right) + n^2$

$a = 25$

$b = 5$

قضیه اصلی:

$n^{\log_5 25} = n^2$

$T(n) = \Theta(n^2 \log n)$

ج) $T(n) = T\left(\frac{3n}{4}\right) + T(n/4) + \underbrace{\frac{1}{2} n^2 (1 - \sin n)}_{O(n^2)}$

$a = 1$ $a = 1$

$b = \frac{4}{3}$ $b = 4$

$n^{\log_{\frac{4}{3}} 1} = n^0 = 1$ $n^0 = 1$

$T(n) = \Theta(n^2)$

تا سلام

Subject:
Date
الف)

4

$\log(n)!$, $\log(\log^*(n))$, $\log^*(\log(n))$, n^9

1 2 3 4

$\log^*(\log(n)) \to k-1$

$\log(\log^*(n)) \to \log k$

5^n, $n2^n$, $100n^2$

5 6 7

Answer

$5^n > n2^n > n^9 > 100n^2 > \log(n)! > \log^*(\log(n)) > \log(\log^*(n))$

5 6 7 4 1 3 2

ب) $n^{n(\log(\log(n)))}$, 10^{100}, $\sum_{i=1}^{n} \frac{n!}{i!}$, n^{100}, $10^9 n$, $\log(n)$, $\log^*(n)$

1

$n^{n(\log(\log(n)))} > n^{100} > 10^9 > \log(n) > \log^*(n) > 10^{100}$ $\cancel{\sum_{i=1}^{n}\frac{n!}{i!}}$

1 3 4 5 6 7

$\sum_{i=1}^{n} \frac{n!}{i!}$

2

$e \approx 2.73$

$e^n = \sum_{i=0} \frac{n^i}{i!} \Rightarrow e^n = \sum_{i=1}^{n} \frac{n!}{i!}$

PAPCO

Subject:
Date

2)
$$\sum_{j=1}^{n}\sum_{i=1}^{j} 1 ,\quad n^{\frac{1}{\log n}} ,\quad \frac{1}{100}\log n ,\quad 10n ,\quad n^4 ,\quad n\log(n) ,\quad n^n$$

3   7   6   5   2   4   1

$$\hookrightarrow n^{\log_n 10}$$
$$= 10\log_n n = \boxed{10^1} = 10$$

$$\hookrightarrow \sum_{j=1}^{n}\left(\sum_{i=1}^{j} 1\right) = \sum_{j=1}^{n} \frac{j(j+1)}{2} = \sum_{j=1}^{n}\left(\frac{j^2}{2}+\frac{j}{2}\right)$$

$$= \frac{n(n+1)(n+2)}{6} \Rightarrow O(n^3)$$

$$n^n > n^4 > \sum_{j=1}^{n}\sum_{i=1}^{j} 1 > n\log(n) > 10n > \frac{1}{100}\log n > n^{\frac{1}{\log n}}$$

5)

Subject:

Date

```
int sum(int n) {
    if (n==1)
        return 1;
    return n + sum(n-1)
}
```

$T(n) = T(n-1) + O(1)$

$+1$

$T(n) = T(n-1) + 1 \Rightarrow T(n-1) = T(n-2) + 1$

$T(n) = T(n-2) + 1 + 1 \Rightarrow T(n) = T(n-n) + n\,\alpha(1)$

$T(n) = T(0) + n$

$T(n) = n$

$\boxed{\Theta(n)}$

Subject:
Date

ب) int Find (int a[], int x)

```
switch (Len(a)) {
    case 0:
        return 0;
    case 1:
        if(x <= a[0])
            return 0;
        return 1;
}

mid = 1 + (len(a) - 1) / 2
if (x <= a[mid - 1])
    return find (a[:mid], x);
return mid + find (a[mid:], x)
```

$O(1)$

$\cancel{(\log n)}$ $O(\log_2 n)$

الگوریتم باینری در
binary search

Subject:
Date

2)

```
void f(int A[]) {
    n = Len(A)
    sq = sqrt(n)
    for(i=1; i<n; i++)
        cout << "*";
    if (n==1)
        return;
    itr = 0;
    while(itr < n) {
        f[A[itr: itr+sq])
        itr += sq;
    }
}
```

$n = Len(A)$, $sq = sqrt(n)$ } $O(1)$

for loop } $O(n)$

if / return / itr = 0 } $O(1)$

n=9 مثال

A[0:3] → [0 1 2], [1 2 3]

A[3:6] ⋮

A[6:9] ⋮

عکس العمل

$$T(n) = \sqrt{n}\,(T(\sqrt{n})) + n$$

$$T(\sqrt{n}) = \sqrt[2^2]{n}\; T(\sqrt[2^2]{n}) + \sqrt[2^2]{n}$$

$$\Rightarrow T(n) = \sqrt{n}\left(\sqrt[2^2]{n}\; T(\sqrt[2^2]{n}) + \sqrt{n}\right) + n$$

$$T(n) = \sqrt{n}\left(\sqrt[2^2]{n}\left(\sqrt[2^3]{n}\; T(\sqrt[2^3]{n}) + \sqrt[2^2]{n}\right) + \sqrt{n}\right) + n$$

$$T(n) = \underbrace{\sqrt{n}\sqrt{n}\sqrt{n}\cdots}_{\log_2 n}\; \underbrace{T(1)}_{O(1)} + (\log_2 n) \times n$$

$$4\left(n^{\frac{1}{2^k}} = 1\right) \qquad 2^k = \log_2 n$$

$$O(n \log_2 n)$$

PAPCO

Subject:
Date

روش دوم

$$T(n) = \sqrt{n}\, T(\sqrt{n}) + n$$

$$\xRightarrow{\div n} \frac{T(n)}{n} = \frac{T(\sqrt{n})}{\sqrt{n}} + 1$$

$$f(n) = f(\sqrt{n}) + 1 \longrightarrow 2^m = n$$

$$\log_2 n = m$$

$$\Rightarrow f(2^m) = f(2^{\frac{m}{2}}) + 1$$

$$\Rightarrow S(m) = S\left(\frac{m}{2}\right) + 1 \Rightarrow O(\log_2 m)$$

تغییر متغیر

$$O(\log \log_2 n)$$

$$O(n \log \log_2 n)$$

جواب روش دوم

سوال ج 5

Subject:
Date

6) در ابتدا باید بررسی کنیم و ببینیم ارزش کد iterative و recursive

چه ارزشی روی هم می‌گذارند و بعد از آن باید به پیچیدگی کد و نسبت آن دو

رابطه‌شون را بنویسیم تا به پیچیدگی برسیم (و سعی می‌کنیم بدترین حالت را بیابیم)

```
int modified B search(arr, target):
    if (len(arr) == 0)
        return -1;
    mid = len(arr) / 2;
    if (arr[mid] == target)
        return mid;
    for (i = mid+1 ; i < len(arr) ; i++)
        if (arr[i] == target)
            return i
    return ..... (arr[: mid], target);
```

$O(1)$

~~$O(\frac{n}{2})$~~
$O(\frac{n}{2})$

در این کد یک آرایه داریم و دنبال یک کلمه با حرف خاص هستیم این کد اینطوری عمل

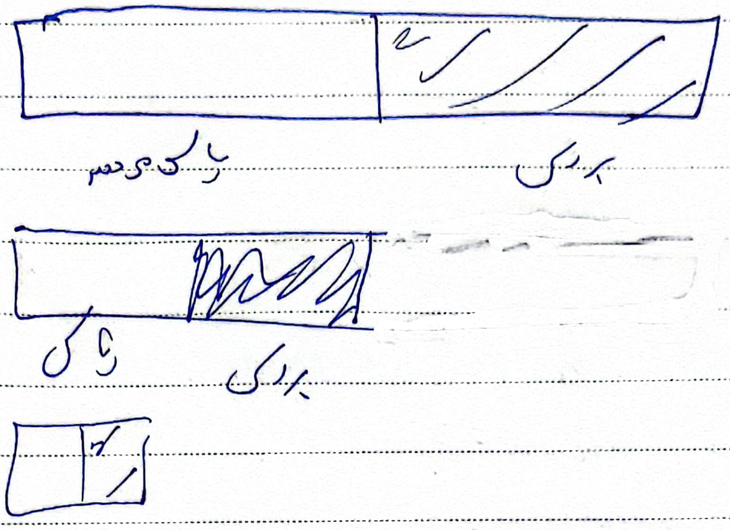
می‌کند که هر وقت صدا زده شود نصف آرایه ما چک و بقیه‌اش پاس داده می‌شود

PAPCO

ادامه سوال 6

یعنی می‌توان گفت در هر مرحله داریم

مرحله 1 $\frac{n}{2}$



مرحله 2 $\frac{n}{4}$

$\frac{n}{8}$

$\vdots$

n

اگر بخواهیم کل بررسی‌ها را بگیریم می‌توان گفت می‌شود

$$\frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \frac{n}{16} + \frac{n}{32} \dots = \sum_{i=0}^{n} \frac{n}{2^i}$$

$O(n)$ که در نهایت می‌شود

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

$f = \frac{n}{2}$ $\quad a = 1$, $b = 2$ $\quad n^{\log_2 1} = 1$

$> O(n)$

$n^{\log_2 1} < \frac{n}{2} \Rightarrow \boxed{O(n)}$ جواب اصلی

Date 7-)

مثال برج هانوی

Honoi(n, A, B, C)::
Honoi(n-1, A, C, B),
Honoi(1, A, B, C),
~~Honoi~~ (n-1, C, B, A)

```
void tower(int n, char A, char B, char C)
{ if (n==1)
      move(A,B);
  else {
      tower(n-1, A, C, B);
      move(A, B);
      tower(n-1, C, B, A);
  };
}
```

برای حل آن داریم

$$T(n) = 2T(n-1)+1 \quad \begin{cases} \to T(n-1) = 2T(n-2)+1 \\ \to T(n-2) = 2T(n-3)+1 \end{cases}$$

$$T(\cancel{n-1}\,n) = 2(2T(n-3)+1)+1 = 2^3 T(n-1) + 2^2 + 2^1 + 1$$

پس در حالت کلی می‌توان نوشت

$$T(n) = 2^k T(n-k) + 2^{(k-1)} + 2^{(k-2)} + \cdots + 2^2 + 2^1 + 1$$

و اگر فرض کنیم $T(1)=1$ پس اگر $k=n-1$ در نظر بگیریم

$$T(n) = 2^{(n-1)} T(1) + 2^{(n-2)} + 2^{(n-3)} + \cdots + 2^2 + 2^1 + 1$$

$$T(n) = O(2^n - 1) \Rightarrow \boxed{T(n) \in O(2^n)}$$

final answer.