1) $T_3 > T_5 > T_2 > T_1 > T_4 > T_6$ $c \geq 0$

a) $T_4 = O(T_6)$

$\ln^2 n = O(\sqrt[3]{n})$

$$\lim_{n\to\infty} \frac{(\ln^2 n)^3}{(\sqrt[3]{n})^3} \Rightarrow \frac{(\ln n)^6}{n}$$ 6 kez pay ve payda türev alınır ise

$$\Rightarrow \frac{360}{n} = 0 \quad n=\infty$$

b) $T_4 = O(T_1)$

$\ln^2 n = O(3n^4+3n^3+1)$

$$\lim_{n\to\infty} \frac{\ln^2 n}{3n^4+3n^3+1} \Rightarrow \frac{2\ln n \ln n}{n(12n^3+6n)} = \frac{2}{n(48n^2+12n)} = 0 \quad n=\infty$$

c) $T_1 = O(T_2)$

$3n^4+3n^3+1 = O(3^n)$

$$\lim_{n\to\infty} \frac{3n^4+3n^3+1}{3^n} \Rightarrow \frac{12n^3+9n^2}{\ln 3 . 3^n} \ldots \Rightarrow \frac{48n^2+18n}{\ln^2 3 . 3^n} = \frac{const}{\infty} = 0$$

d) $T_2 = O(T_5)$

$3^n = O(4^n)$

$$\lim_{n\to\infty} \frac{3^n}{4^n} = \frac{\ln 3 . 3^n}{\ln 4 . 4^n} \Rightarrow 0$$

$n=\infty$ da payda daha hızlı büyüyor

e) $T_5 = O(T_3)$

buradan $(n-2)!$ 'ın türevini alırken tanımsız olacağı için stirling'in formülünü uygulamalıyız $n! \cong \sqrt{2\pi}.n^{(n+1/2)}.e^{-n}$

$4^n = O((n-2)!)$

$$\lim_{n\to\infty} \frac{4^n}{(n-2)!} \Rightarrow \frac{4^n}{\sqrt{2\pi}.(n-2)^{(n-2+\frac{1}{2})}.e^{-(n-2)}} = \frac{(4^n . e^{n-2})'}{(\sqrt{2\pi}.(n-2)^{-3/2})'} = \frac{(\ln 4+1).e^{n-2}.4^n}{\frac{-3\sqrt{15\pi}}{10(n-2)^{5/2}}} = \infty$$

$n=\infty$ konur ise

2) Bir array içindeki en büyük ve en küçük elemanları bulup bunların ortalamasına en yakın elemanı return eder.

fruits: array

plum: en başta max sayı ile set edilmiştir, en sonda ise arraydeki min. sayı oluyor

watermelon: arraydeki en büyük sayıyı bulmak için kullanılıyor

orange: büyük ve küçük sayıların ortalamasına en yakın sayı

orangeTime: while döngüsünden çıkmak için kullanılan flag.

```
def delicious(fruits)
  plum = sys max
  watermelon=0
  orange=0
  orangeTime=false
  while not orangeTime
    for fruit in fruits
      if fruit>watermelon
        watermelon=fruit
      if fruit<plum
        plum=fruit
        break
    else
      orangeTime=True
  for fruit in fruits
    if (abs ... )
      orange = fruit
  return orange
```

Worst Case

Array büyükten küçüğe sıralı ise ve en büyük eleman en başta ise; n^2'de çalışır. $O(n^2)$

Best Case

Although for; her durumda n kere dönecek ve array küçükten büyüğe sıralı ve en küçük eleman en başta ise; n'de çalışır $\Omega(n)$

Average Case

Bu durumda ise worst ve best case'in ortalaması yani $\frac{n+n^2}{2}$'de çalışır.

kısaca n^2 denebilir. $\Theta(n^2)$

3) a)

```
int sum=0;
for (int i=0; i<n-1 ;++i)
    sum += sqrt(sqrt(i)+1);
```

$\Theta(n^5)$

$1+4+\dots+n^4 < n.n^4$

b) $\log 2^2 + \log 3^2 + \dots + \log(n-1)^2 < 2(n-2).\log(n-1)$

$\Theta(n \log n)$

c) $+2.2^0 + 3.2^1 + 4.2^2 + 8.2^3 + \dots + (n+1).2^{n-1} < n(n+1).2^{n-1}$

$\Theta(n.2^n)$

d)

```
int sum=0;
for (int i=0; i<n-1 ;++i)
    for(int j=0; j<i-1 ;++j)
        sum += (i+j);
```

iç döngü

$$\sum_{i=0}^{n-1}\Big(\underbrace{\underbrace{\sum_{j=0}^{i-1} i}_{i} + \underbrace{\sum_{j=0}^{i-1} j}_{i(i-1)}}_{\text{Buradan } i^2 \text{ gelir}}\Big) = \Theta(n^3)$$

4) iç for döngüsü, dıştakine bağlı olduğu için beraber düşünülmeli, üst döngünün başlangıç değeri n olduğu için ve iç döngü her seferinde $\frac{1}{2}$ oranında daha az döndüğü için; dış döngüdeki i=n iken kod bloğunun çalışma sayısı, sırasıyla;

$$n, \frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \ldots, 1$$

üst for'da integer bölmesi yapıldığı için en sonda 1'e kadar gelinecek

$$n\left(\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \ldots + \frac{1}{2^{m-1}}\right)$$

Summation;

→ burada m, iç döngü'nün iterasyon sayısı kadardır

$$\sum_{i=0}^{m-1} n\left(\frac{1}{2}\right)^i = O(n)$$

5) a) $n^3 \in O(3^{2n})$ $c \geq 0$

$$\lim_{n \to \infty} \frac{n^3}{3^{2n}} \Rightarrow \frac{3n^2}{\ln 9 \cdot 9^n} \ldots$$

türev aldıkça pay const olacak, payda n'li

$$= \frac{const}{\infty} = 0$$ ✓

b) $n \in o(\log\log n)$ $c = 0$

$$\lim_{n \to \infty} \frac{n}{\log\log n} \Rightarrow \frac{1}{\frac{1}{n \cdot \log n (\ln 2)^2}} = \infty$$ ✗

c) $n^2 \log^2 n \in O(n!)$ n! yerine stirling varsayımını uygularsak

$$\lim_{n \to \infty} \frac{n^2\log^2 n}{n!} \Rightarrow \frac{(n^2\log^2 n)'}{(\sqrt{2\pi} \cdot n^{(n+1/2)} \cdot e^{-n})'} = \frac{2\cancel{n\ln n}(\ln n+1) \cdot 10}{\sqrt{2\pi}\, n^{n-\frac{1}{2}} \cdot e^{-n}(2\cancel{n\ln n}+1)}$$

Tekrar Türev

$$= \frac{const}{n'li\ değer} \Rightarrow \frac{const}{\infty} = 0$$ ✓

n=∞ konursa

d) $\sqrt{10n^2+7n+3} \in \Theta(n)$ $c > 0$

$$\lim_{n \to \infty} \frac{\left(\left(\sqrt{10n^2+7n+3}\right)^2\right)'}{\left((n)^2\right)'} \Rightarrow \frac{20n+7}{2n} = \frac{20}{2} = 10$$ ✓