1) $f_{12}(n) < f_{14}(n) < f_{11}(n) < f_{10}(n) < f_2(n) < f_{13}(n) < f_3(n) < f_4(n) < f_1(n) < f_8(n) < f_9(n) < f_7(n) < f_5(n) < f_6(n)$

2) a) $\frac{n^3-n}{3}$ b) $r \leftarrow r+1$ c) $T(n) = O(n^3)$

```
r <- 0
for i=1 to n                ---> n
  for j=i+1 to n            ---> n-i-1
    for k=i+j-1 to n        ---> n-i-j+1
      r <- r+1
return r
```

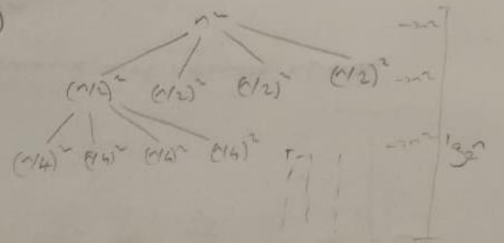
$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=i+j-1}^{n} 1 = \sum_{i=1}^{n}\sum_{j=i+1}^{n}(n-i-j+2) \Rightarrow \sum_{i=1}^{n}\sum_{j=i+1}^{n}(n-i+2) - \sum_{i=1}^{n}\sum_{j=i+1}^{n} j$$

$$= \sum_{i=1}^{n}(n-i)\cdot(n-i+2) - \sum_{i=1}^{n}(n-i) = \boxed{\frac{n^3-n}{3}}$$

$\boxed{\text{Basic operation } r \leftarrow r+1}$

$\boxed{T(n) = O(n^3)}$

3)



$T(n) = n^2 \log_2 n$

$\boxed{T(n) = n^2 \log_2 n = O(n^2 \cdot \log_2 n)}$

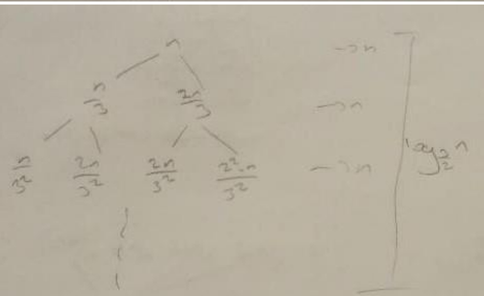
$n/2^h = 1 \Rightarrow 2^h = n$

$\log_2 2^h = \log_2 n$

$h \log_2 2 = \log_2 n$

$\boxed{h = \log_2 n}$

4)



$T(n) = n \cdot \log_{\frac{3}{2}} n$

$T(n) = n \log n$

$T(n) = O(n \log n)$

we continue with rightmost sub-tree

$$\frac{n}{\left(\frac{3}{2}\right)^h} = 1$$

$n = \left(\frac{3}{2}\right)^h$

$\log n = h \log \frac{3}{2}$

$\boxed{\log_{\frac{3}{2}} n = h}$

5) $r \leftarrow 0$

for $i=1$ to $n-2$ $\longrightarrow (n-2-1)=n-3$

for $j=i+1$ to n $\longrightarrow n-i-1$

if $a_j \neq a_j$

return false $\longrightarrow 1$

return true

$$\sum_{i=1}^{n-2} \sum_{j=i+1}^{n} 1$$

$$= \sum_{i=1}^{n-2} (n-i-1)+1$$

$$= \sum_{i=1}^{n-2} (n-i)$$

$$= (n-1) + (n-2) + (n-3) - - - - + 2 + 1 - 1$$

$$= \frac{(n-1).(n)}{2} - 1$$

$$= \boxed{\frac{n^2-n-2}{2}}$$

Basic operation is if $a_{ij} \neq a_j$

$T(n) = O(n^2)$

6) $a=2$

$b=2$

$k=1$

$p=1$

clearly $a = b^k$

$2=2$

we follow case 2

Since $p > -1$

$$T(n) = O(n^{\log_b a} . \log^{p+1} n)$$

$$= \boxed{O(n \log^2 n}$$

7) $a=3$

$b=3$

$k=\frac{1}{2}$

$p=0$

clearly $a > b^k$

we follow case 1

Then we have

$$T(n) = O(n^{\log_b a})$$

$$= \boxed{O(n)}$$

8) This algorithm takes sequence of integers and returns the min of the sequence

```
SAMSUN(a_i, a_{i+1}, ..., a_j)
input: a sequence of integers

if i=j
  return a_i
else
  mid ← ⌊(i+j)/2⌋
  temp1 ← SAMSUN(a_i, ..., a_mid)
  temp2 ← SAMSUN(a_{mid+1}, ..., a_j)
  if temp1 ≤ temp2
      return temp1
  else
    return temp2
```

$$T(n) = 2T\left(\frac{n}{2}\right) + 1$$

Build solution

$T(n) = 2T\left(\frac{n}{2}\right) + 1$

$= 2\left[2T\left(\frac{n}{2^2}\right) + 1\right] + 1 \Rightarrow 2^2 T\left(\frac{n}{2^2}\right) + 2 + 1$

$= 2^2\left[2T\left(\frac{n}{2^3}\right) + 1\right] + 1 + 1 \Rightarrow 2^3 T\left(\frac{n}{2^3}\right) + 4 + 2 + 1$

Expand Scratch

$T(n/2) = 2T\left(\frac{n}{2^2}\right) + 1$

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

$= 2^i T\left(\frac{n}{2^i}\right) + 2^{i+1} - 1 \quad T(1) = 1 \quad \frac{n}{2^i} = 1 \quad n = 2^i \quad \log n = i \log 2$

$\log_2 n = i$

$\sum_{i=1}^{k} 2^i = 2^{k+1} - 2$

$= 2^{\log_2 n} T(1) + 2^{\log_2 n} - 1 = n + n - 1$

$= T(n) = O(n)$