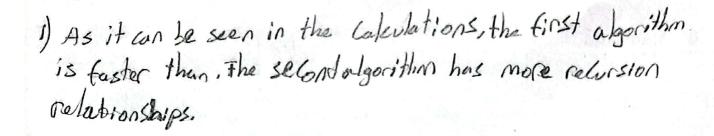
(a) Algorithm 
$$alg1(L[0..n-1])$$
if  $(n=1)$  return  $L[0]$   $O(I)$ 
else

tmp =  $alg1(L[0..n-2])$   $T(n-I)$ 
if  $(tmp <= L[n-1])$  return tmp  $I(I)$ 
else return  $I(I)$ 



The operations that can be done are minimized, it is not possible to find the result without going through all the coefficients. So this is the best time complexity.

1) 
$$T(n) = T(n/1) + o(1)$$
  
 $T(n/2) = T(n/2) + o(1)$   
 $T(n/2) = T(n/3) + o(1)$   
 $T(n) = T(n) + o(1)$   
 $T(n) = T(n) + (n-1) \cdot o(1)$   
 $T(n) = o(n)$ 

Okan Joren 1801042662

1801047662 1801047662

for i=0 to len(letters)): ) 
$$O(n)$$

for i=0 to len(letters)): )  $O(n)$ 

if (letters[i]!= start)  $O(n)$ 

break

else

if (letters[j]==end)

Count=count+1

return count

return min.

$$f(n) = \sum_{i=0}^{n} \sum_{j=1}^{n} \varphi(n)$$

$$f(n) = \sum_{i=0}^{n} \varphi(n) = \varphi(n^{2})$$

$$f(n) = \varphi(n^{2})$$

Okun Jorun 1801042662 mostprofitable Cluster (brench Nome, branch No) Maxz= # on for i to to ten (branch Name) )O(1) for j=i to len(branch Name) ) 1 temp. profit=0 ) (1) for k= ; to j+1 temp-profit = temp-profit + branchwo[] temp-cluster = temp. cluster + branch Nume[k] if temp-profit > maxel max = temp\_profit max2 = temp\_cluster

Print (max 2)

$$\sum_{i=1}^{j-1} \sum_{i=1}^{j-1} \sum_{i=1}^{j-1}$$

Jour Jours 201042662