given string is a palindrome? b a b a d

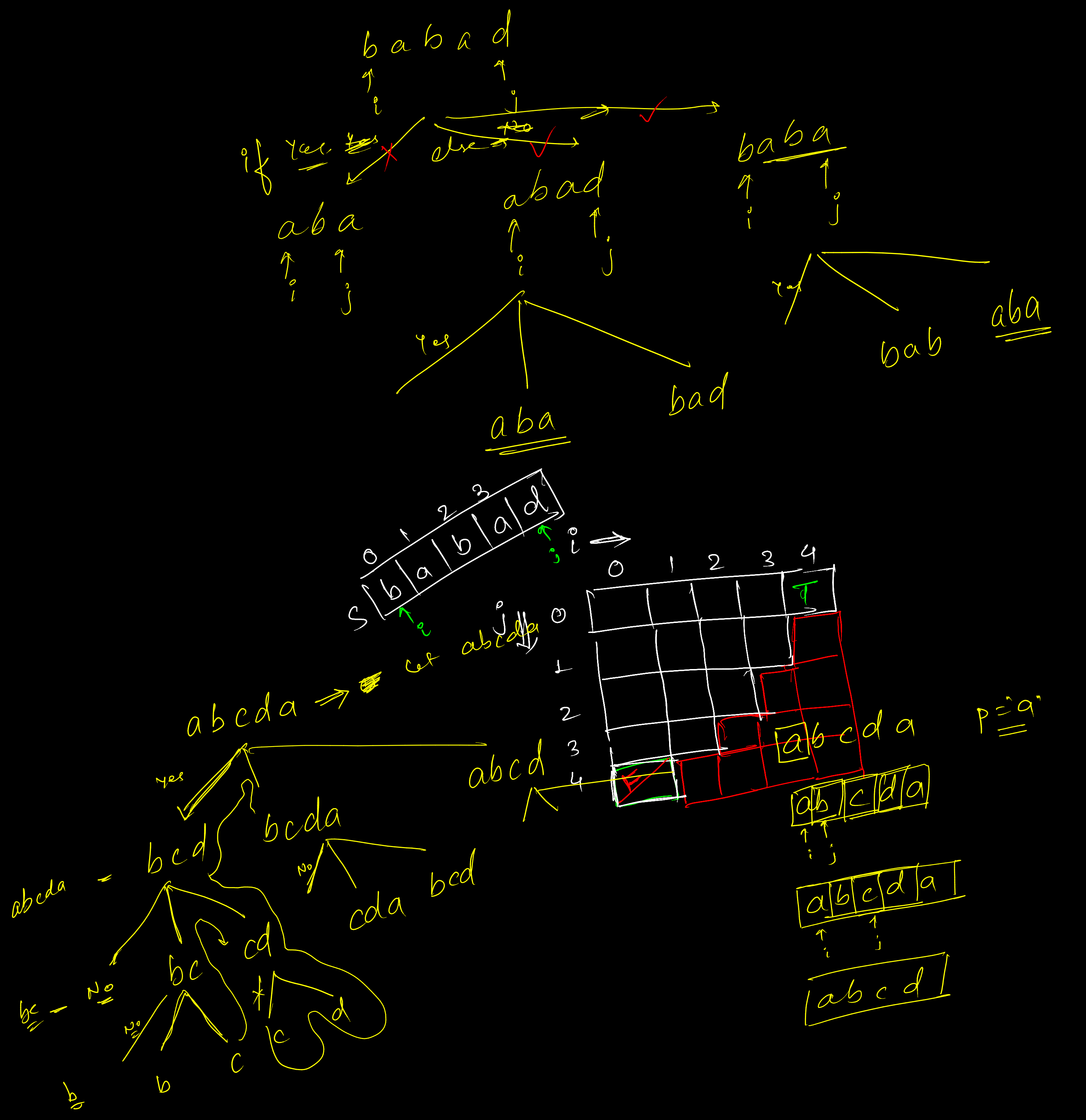
↑ i ↑ j

while (i<j)

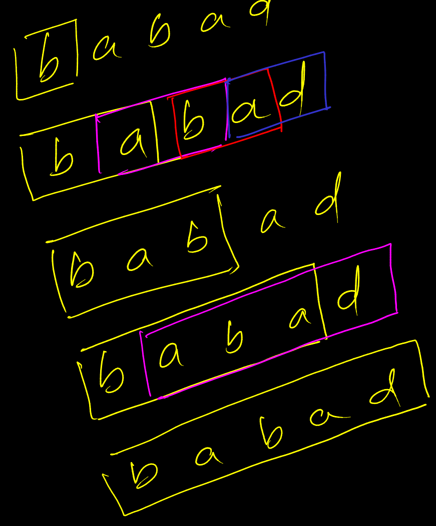
→ check.

Need-ques.

↳ How many palindromes a string can have.



b a b a d



⇒ p = b

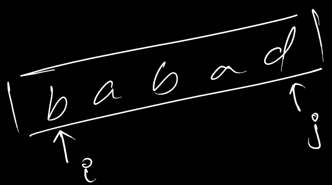
⇒ No change o/p

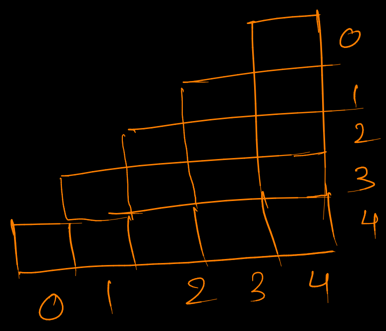
⇒ p = bab

⇒ No change

⇒ No change.

Let's reverse this process





string is immutable ? in cpp

plind checker(&string , i, j ; value store table)
{
if (i<j string[i] = string[j])
return palind(

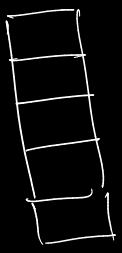
else

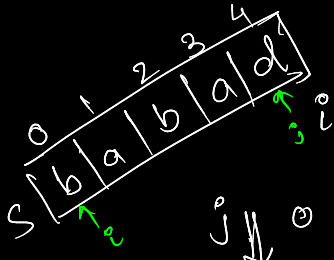
string S = babad

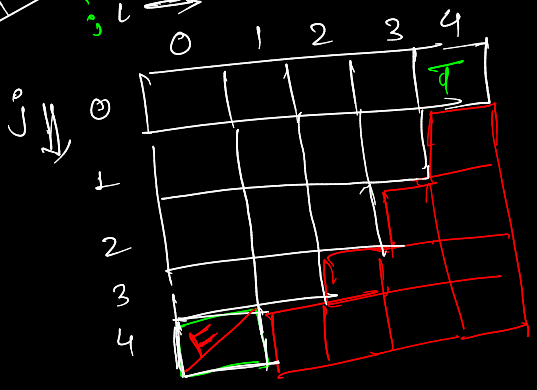
int L = S.length()

base cond"

```
if(!L)
    return "";
if(L==1)
    return s;
if(L==2 && S[1]==S[0])
    return s;
else return S[0];
```








```
pchacker(i, j)
{
    if(i<j)
        if(S[i]==S[j])

    else
```

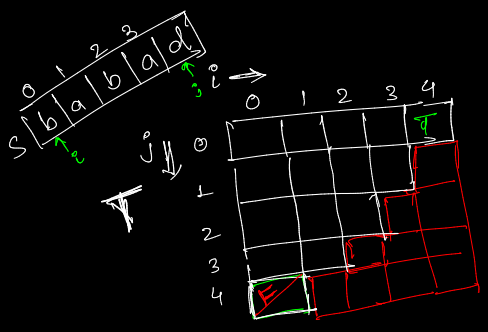
S = 0 1 2 3 4
b a b a d

f(0, 4)

if(S[i]=S[j])

return: f(1, 3)

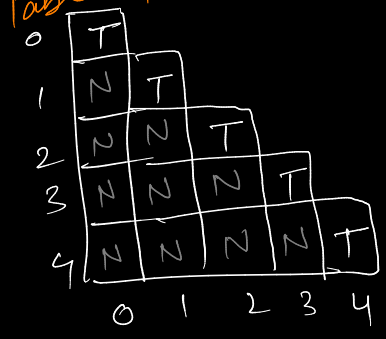
$$f(i,j)\begin{cases} T[i][j]=T \\ f(i+1, j-1) & \text{if } (i==j) \text{ if } S[i]=S[j] \\ f(i+1, j) \\ f(i, j-1) & \text{otherwise} \end{cases}$$



```
if (s.len() == 0)
  return "";
checker( i = 0 , j = s.len() - 1)
```

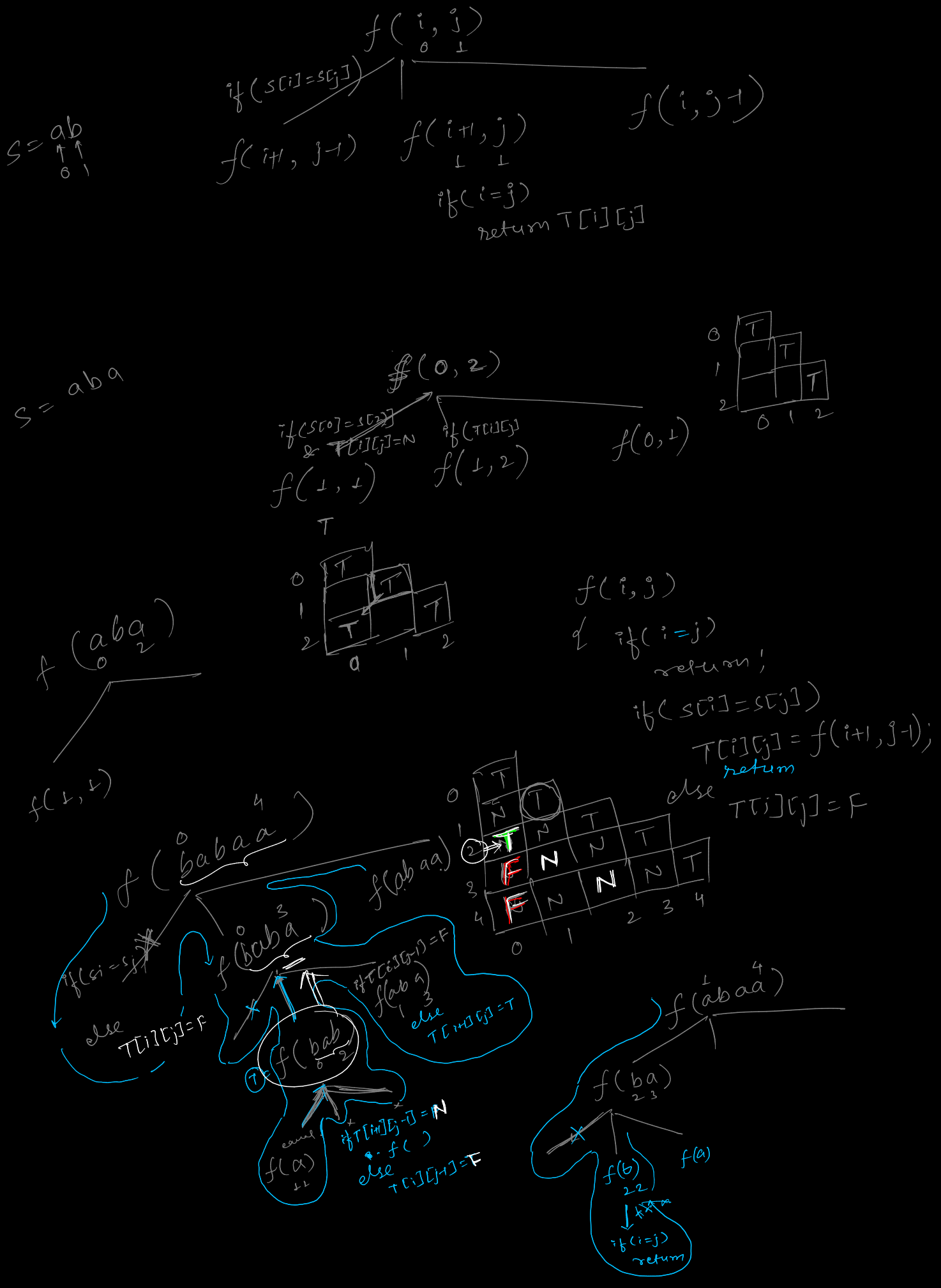
babad\0
↑ i ↑ j

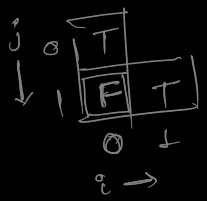
Table : T

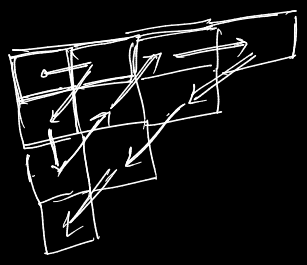


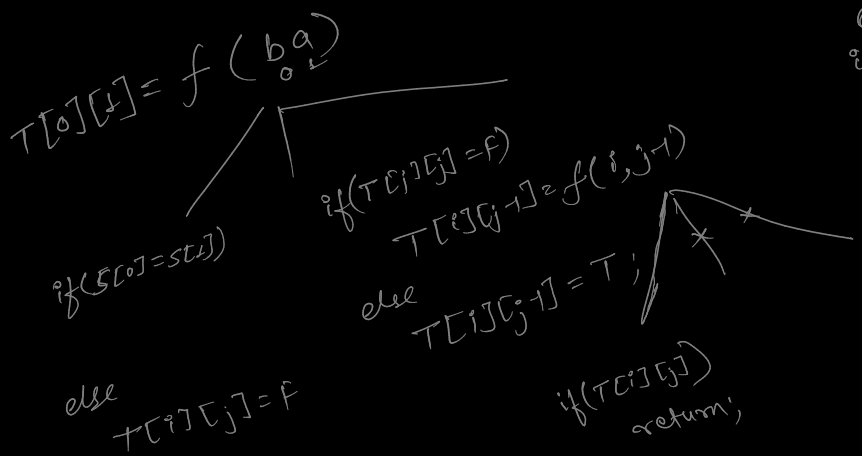

```
checker(      )
{   if ( T[i][j] ≠ N )
        s[j+1] = "\0";
        return i;
```

s = a
↑ ↑
i j


f(i, j)
if(s[i]=s[j])
f(i+1, j-1)
f(i+1, j)
f(i, j-1)
s = ab
if(i=j)
return T[i][j]
s = aba
f(0, 2)
f(1, 1)
f(1, 2)
f(0, 1)
T
f(aba)
f(1, 1)
f(i, j)
if(i=j)
return;
if(s[i]=s[j])
T[i][j] = f(i+1, j-1);
return
else
T[i][j] = F
f(baaba)
f(abaa)
f(baba)
f(bab)
f(a)
f(aba)
f(ba)
f(b)
f(a)
if(i=j)
return
else
T[i][j]=F
if T[i+1][j-1]=F
f(aba)
else
T[i+1][j]=T
if T[i+1][j-1]=N
& f( )
else
T[i][j-1]=F







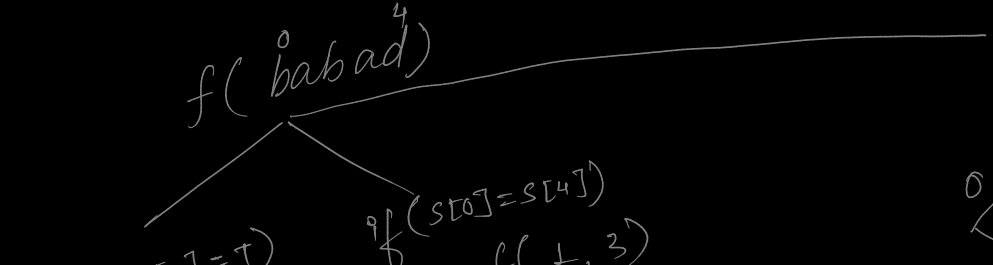

```
check(i, j)
{
    if (T[i][j])
        return;
```

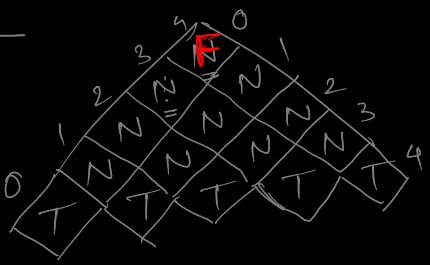
T
return;

```
check(0,0)
{
    return;
```

	0	1	2
-1		T	
F	N		

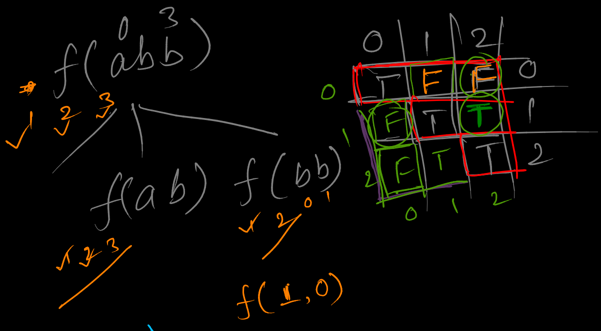
```
for(i=0; i<n; i++)
    for(j=n; j>0; j--)
```



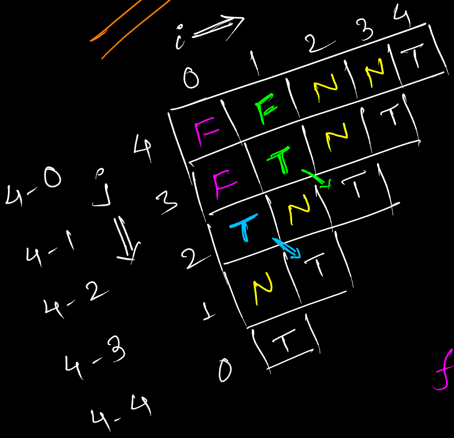


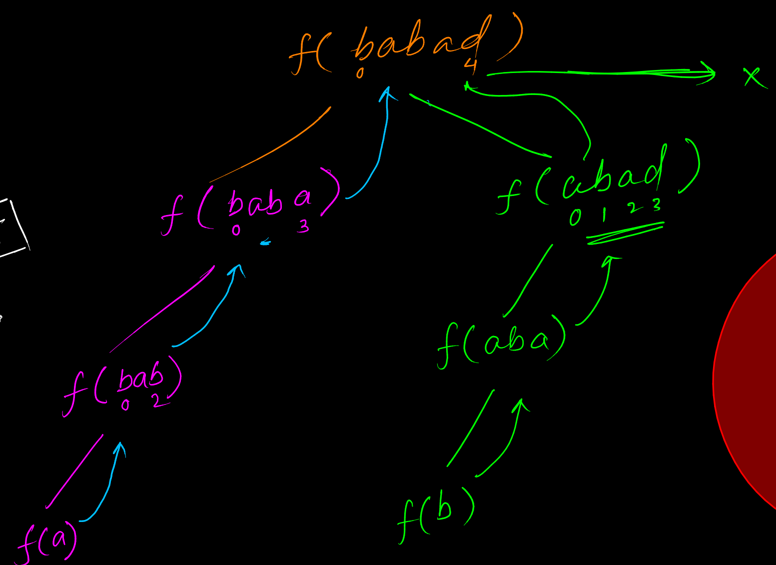
[0][4]
[0][3]
[1][4]
[2][4]
[1][3]

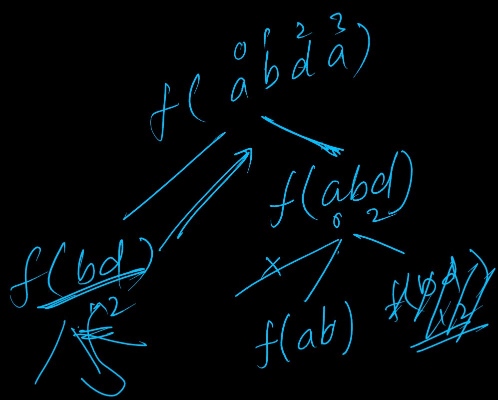
```
checker(i, j)
{
① if (T[i][j])      T or F then return
                    else compute
      return;

② if (s[i] = s[j])
      if(i+1 ≠ j)
      checker(i+1, j-1);  →  if(T[i+1][j-1])
                                 T[i][j] = T
                             else goto Label;
      else T[i][j] = T return;

③ else
   Label: T[i][j] = F
      checker(i, j-1)

      if(T[i][j-1] = F)
      checker(i+1, j)
```

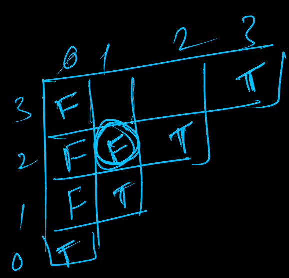


Let's Analyze our code







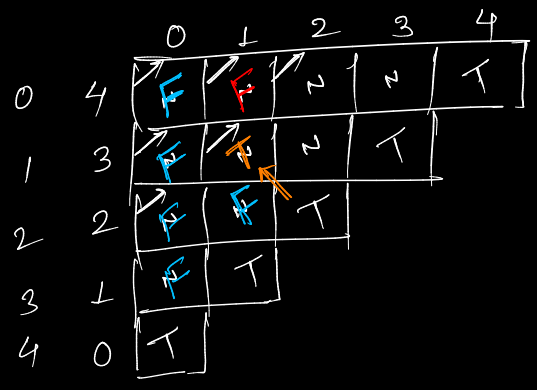


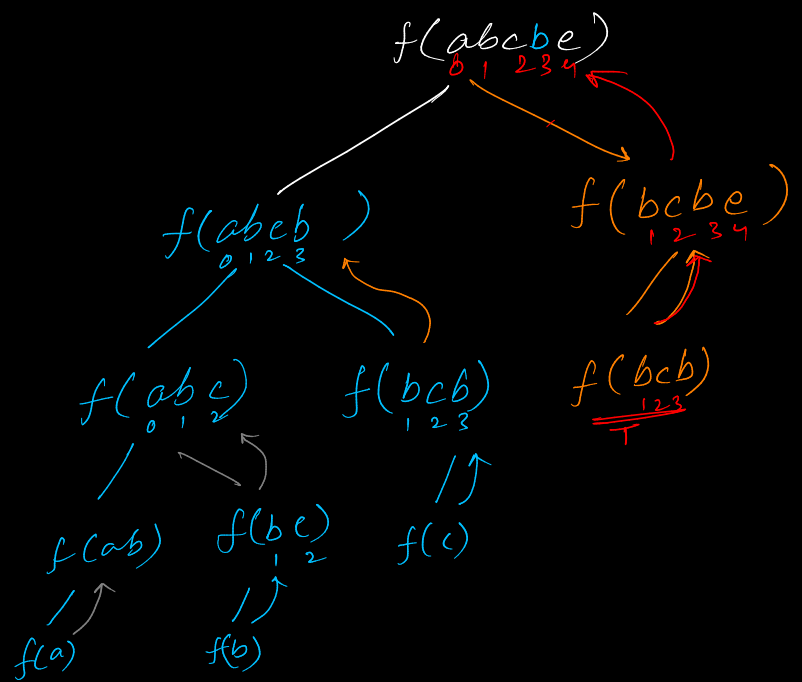
Final Analysis

TC = $O(2^n)$; SC = $O(1)$

to TC = $O(n^2)$; SC = $O(n^2)$

I learned a lot.






```
checker(i, j)
{
  if (T[i][j])        T or F mai return
                      else compute
    return;

  if (s[i] = s[j])
    if(i+1 != j)
      checker(i+1, j-1);
    else T[i][j] = T return;
  else
    { label: T[i][j] = F
      checker(i, j-1)

      if(T[i][j-1] = F)
      checker(i+1, j)
    }
```

if(T[i+1][j-1])
T[i][j] = T
else goto label;