

Count Palindromic Substrings

abc cbc

✓ a

ab

abc

abcc

abccb

abc cbc

✓ b

bc

bcc

✓ bccb

bcc bc

✓ c

✓ cc

ccb

ccbc

✓ c

cb

✓ cbc

✓ b ✓ c

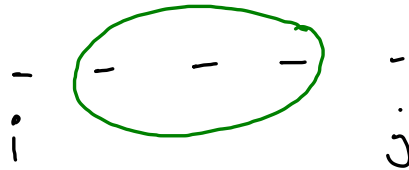
bc

a b c c b c

i \rightarrow st.

j \rightarrow et.

$(i+1, j-1)$



① $ch(i) == ch(j)$

83

$isPal(i+1, j-1) \rightarrow true$

abc cbc

et \rightarrow

count \rightarrow no. of True's
palindromic
Substrings.

$i \rightarrow st$

$j \rightarrow et$

$(i, j) \rightarrow (i+1, j-1)$

[row traversal \cup (5 to 0)
col traversal \cup (0 to 5)
diagonal traversal

diagonal traversal (logical
practically)

st



	a_0	b_1	c_2	c_3	b_4	c_5
a_0	T	F	F	F	F	F
b_1	X	T	F	F	T	F
c_2	X	X	T	T	F	F
c_3	X	X	X	T	F	T
b_4	X	X	X	X	T	F
c_5	X	X	X	X	X	T

Longest Palindromic Substring

et \rightarrow

st



	a_0	b_1	c_2	c_3	b_4	c_5
a_0	T	F	F	F	F	F
b_1	X	T	F	F	T	F
c_2	X	X	T	T	F	F
c_3	X	X	X	T	F	T
b_4	X	X	X	X	T	F
c_5	X	X	X	X	X	T

the last 'true' while
travelling will be the
ans.

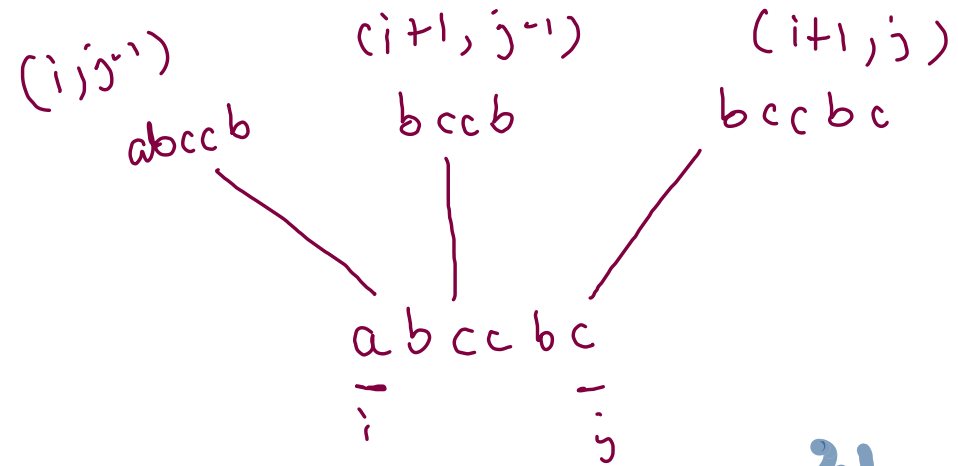
abc cbc

recursion

et \rightarrow

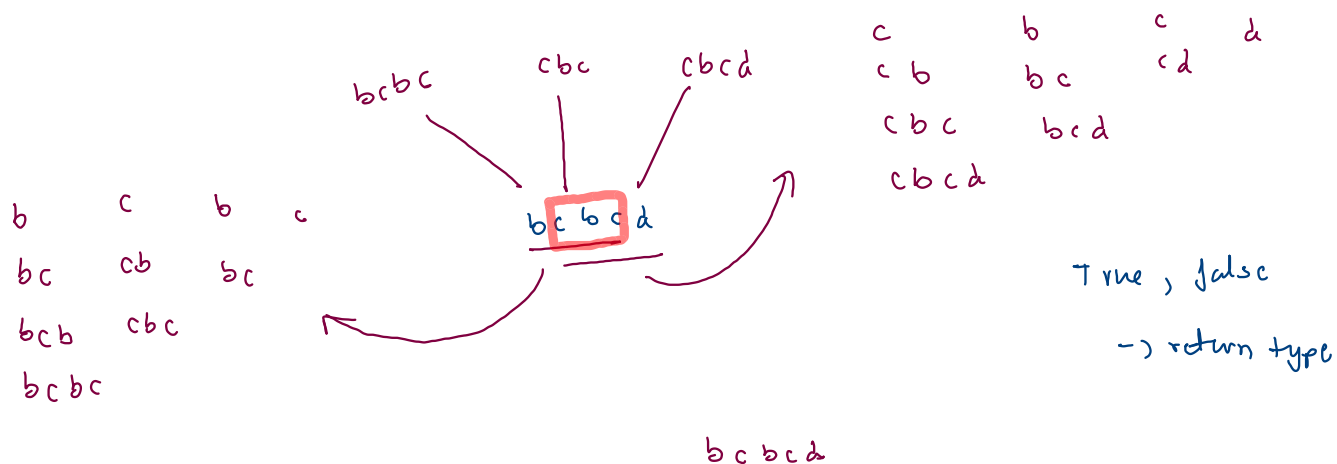
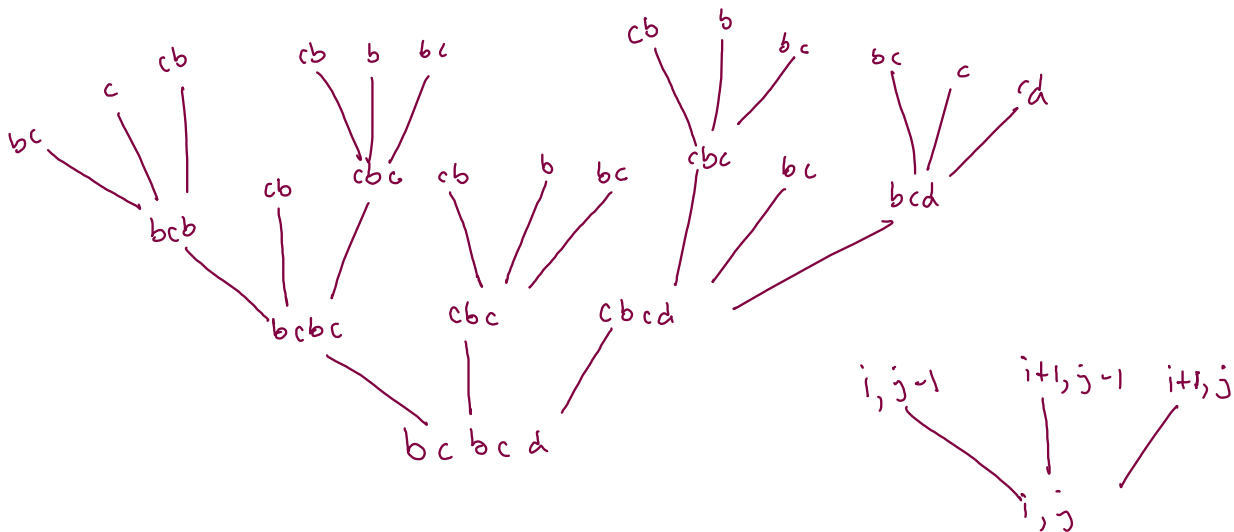
	a_0	b_1	c_2	c_3	b_4	c_5
a_0	T	F	F	F	F	F
b_1	X	T	F	F	T	F
c_2	X	X	T	T	F	F
c_3	X	X	X	T	F	T
b_4	X	X	X	X	T	F
c_5	X	X	X	X	X	T

st
 \downarrow



21

b c b c d



Print All Paths With Minimum Cost

DFS / BFS

```
014282
436504
124146
207322
315924
270851
```

	0	1	2	3	4	5
0	0	1	4	2	8	2
1	4	3	6	5	0	4
2	1	2	4	1	4	6
3	2	0	7	5	2	2
4	3	1	5	9	2	4
5	2	7	0	8	5	1

cost

	0	1	2	3	4	5
0	23	23	24	20	21	19
1	24	22	23	18	13	17
2	20	19	17	13	13	13
3	21	19	19	12	9	7
4	23	20	19	16	7	5
5	23	21	14	14	6	1

dp

dp[i][j] ->

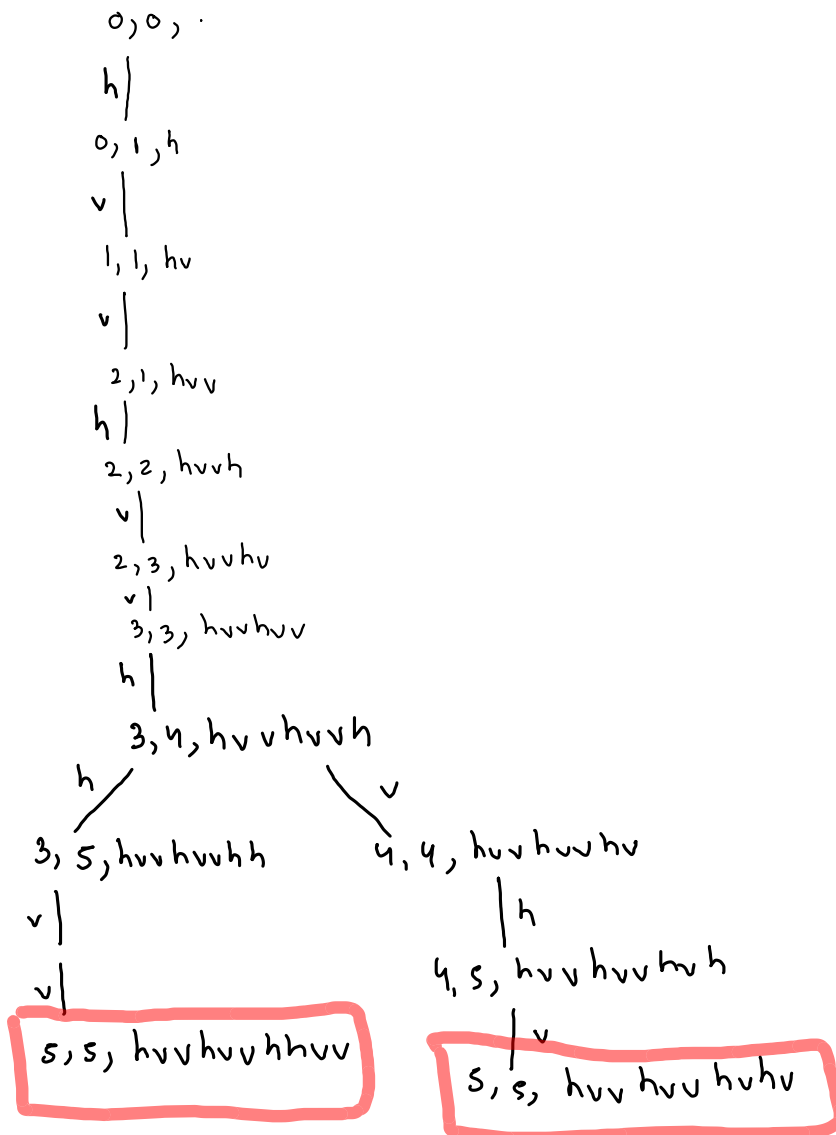
i, j to m, m-1
min cost



DFS

	0	1	2	3	4	5
0	23	23	24	20	21	19
1	24	22	23	18	13	17
2	20	19	17	13	13	13
3	21	19	19	12	9	7
4	23	20	19	16	7	5
5	23	21	14	14	6	1

dp



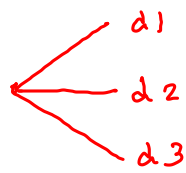
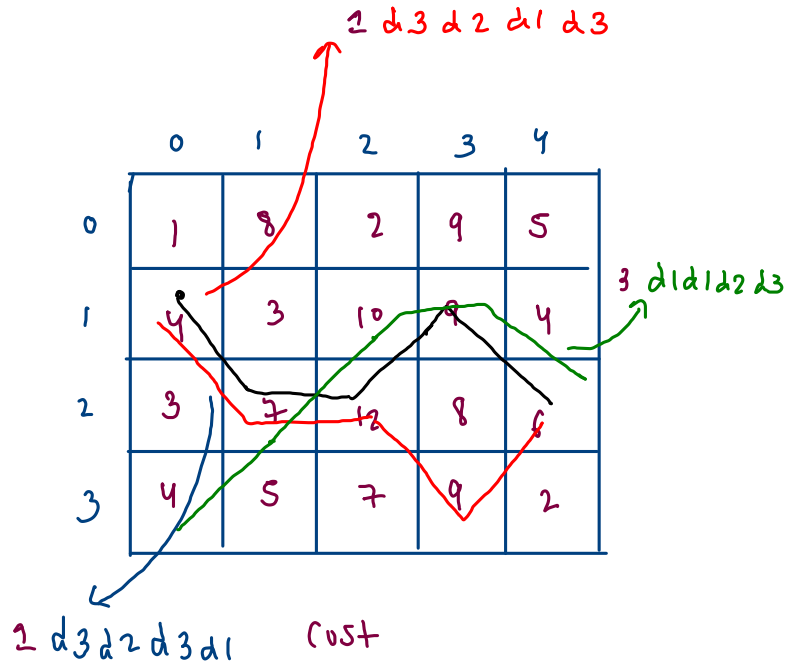
	0	1	2	3	4	5
0	23	23	24	20	21	19
1	24	22	23	18	13	17
2	20	19	17	13	13	13
3	21	19	19	12	9	7
4	23	20	19	16	7	5
5	23	21	14	14	6	2

dp

BFS
DFS } same

java (iterative soln has
a upper hand
always)

Print All Paths With Maximum Gold



	0	1	2	3	4
0	34	33	17	14	5
1	38	30	25	15	4
2	37	34	27	14	6
3	38	32	22	15	2

$dp[i][j]$;
max gold
collected if
we start digging
from i, j towards last col.

