

$s = a b d b e b c a$

$\Rightarrow S$

$a b d b a$
 $a b b b a$
 $a b e b a$

DP table for the string $s = a b d b e b c a$ (length 8, $n=8$).

Indices: i (row) and j (column) range from 0 to 7.

Values in the table represent the number of palindromic substrings ending at (i, j) .

	a	b	d	b	e	b	c	a
a	1	1	1	3	3	3	3	5
b	0	1	1	3	3	3	3	5
d	0	0	1	1	1	3	3	5
b	0	0	0	1	1	3	3	5
e	0	0	0	0	1	1	1	4
b	0	0	0	0	0	1	1	4
c	0	0	0	0	0	0	1	4
a	0	0	0	0	0	0	0	1

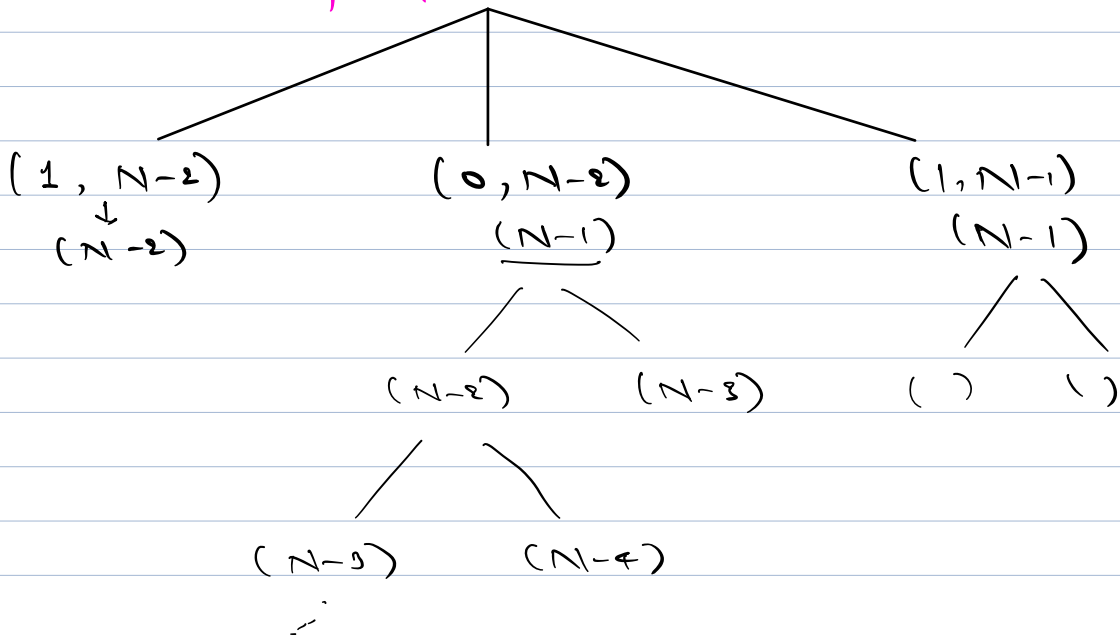
Annotations:

- $i=0$ (row index)
- $j=7$ (column index)
- $i=j$ (diagonal elements)
- substrings of size 1 (diagonal elements)
- $(j-i)$ (length of substring)

$a b d b$

$$[i, j] \Rightarrow \text{Length} = (j - i + 1)$$

$$\text{llps}(0, N-1) \Rightarrow \underline{N}$$



①

Code

```
DP[N][N] = 0;
```

```
for (i = 0; i < N; i++) {
```

```
    DP[i][i] = 1;
```

```
}
```

```
for (l = 2; l <= N; l++) {
```

```
    i = 0;
```

```
    j = l - 1;
```

```
(l = j - i + 1)
```

```
j = l + i - 1
```

```
    while (j < N) {
```

```
        if (s[i] == s[j]) {
```

```
            DP[i][j] = 2 + DP[i+1][j-1];
```

```
        }
```

```
        else {
```

```
            DP[i][j] = max(DP[i+1][j],  
                           DP[i][j-1]);
```

```
        }
```

```
        i++;
```

```
        j++;
```

```
    }
```

```
}
```

$$T.C. = O(N^2)$$

$$S.C. = O(N^2)$$

Q

Given a string of size N .

$\forall i, j$ tell whether the substring from i to j is a palindrome. ??

S: 0 1 2 3 4 5 6 7 $(N=8)$
 a b b c c c b a

O/P ?? $\Rightarrow N^2$ boolean matrix.

$DP[N][N] = 0;$

for ($i=0$; $i < N$; $i++$) {

$O(N^2)$ [

for ($j=i$; $j < N$; $j++$) {

if (check(i, j)) { $O(N)$

$DP[i][j] = 1;$

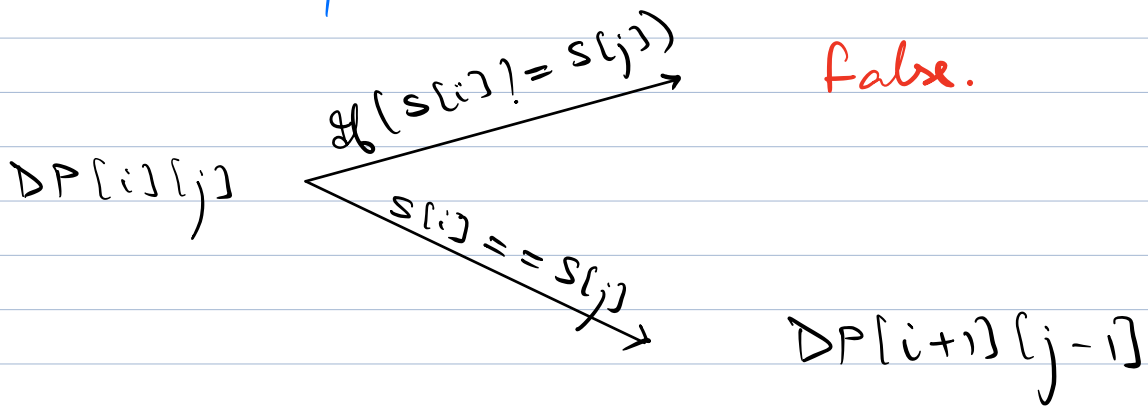
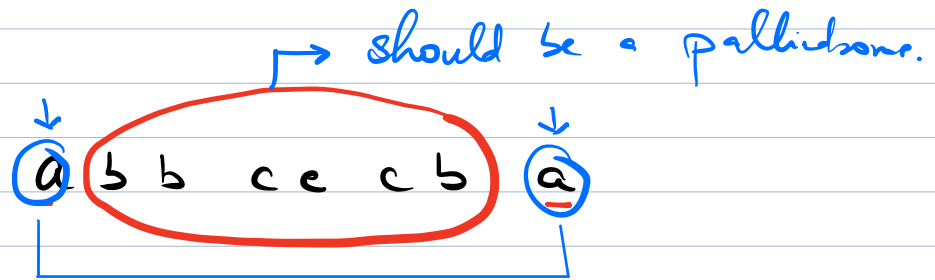
}

}

}

check if substring from i to j is a palindrome or not. ??

$$T.C. = O(N^3)$$



Code → H.W.



Given a string of size N.

find the length of the longest palindromic substring.

$$O(N^2) =$$

$$\text{Length} = (j-i+1)$$

Iterate over the prev goes sol's

$\forall i, j$ where $DP[i][j] = 1$

$(j-i+1) \leq$ maintain a max.

$$\underline{O(N^2)}$$



Word Break

Amazon
FB
Microsoft
GS
Adobe
Google
MS
...

Given a dictionary (list of words)

Given a string (a-z, no spaces)

Check if it is possible to break down the string into valid words from the dictionary.

am | man | go

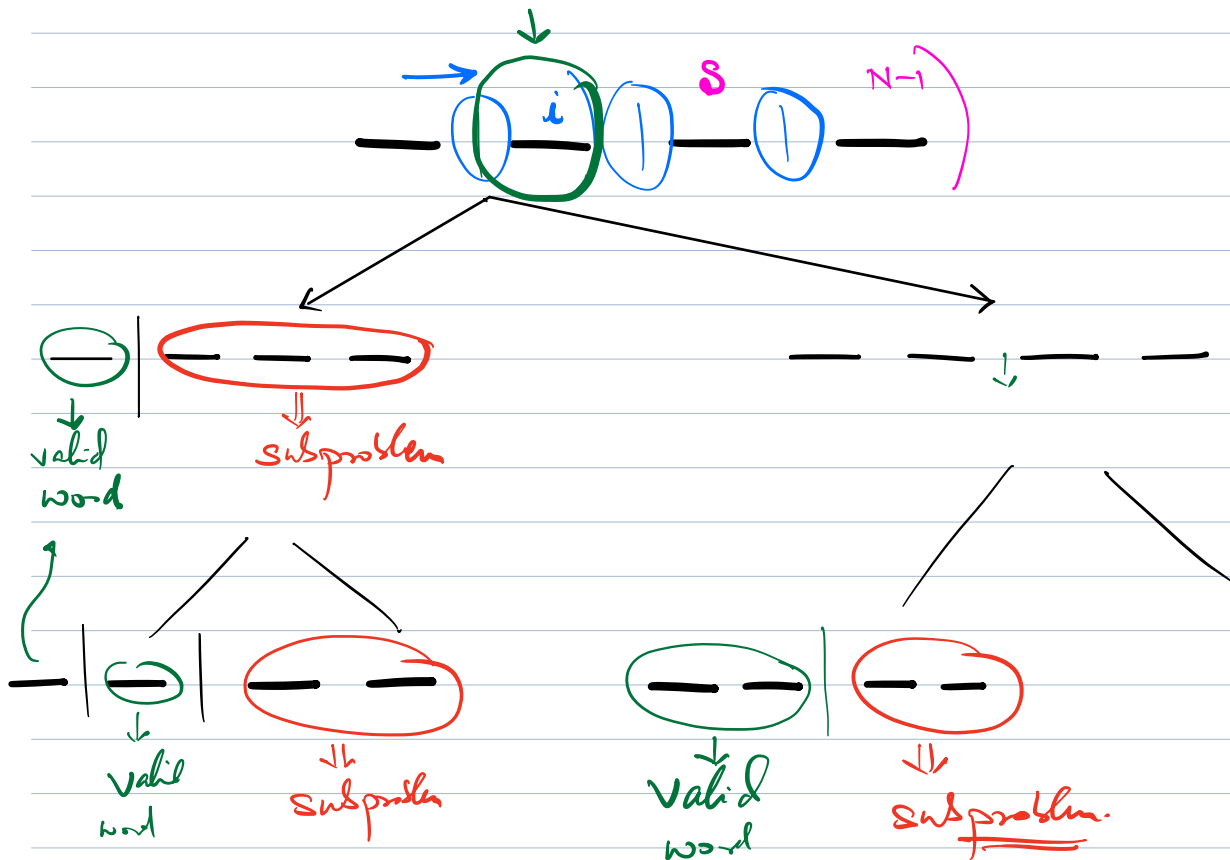
Eg: dict: { like, i, man, go, mangoes }

1) i | like | mangoes ✓

2) i | like | go | ~~es~~

3) i | like | i | man | mangoes | go ✓

4) mangoes ✓



Word(s) → Returns true if it's possible to break the string `[s, N-1]` into valid words from the dictionary.

0, N-1

boolean wordBreak (s, e, str) {

// if (s == str.length()) {
 return True;

for (i = s; i <= e; i++) {

if (isValidWord(s, i, str) && wordBreak(i+1, e, str))
 return True;

return false;

}

H.W.

(DP solution)

↓
Iterative

Pallindrome Partition