

**NAMA : Fajar Satria**  
**NRP : 05111940000083**  
**Kelas : PAA B**

## PERMASALAHAN – LNCPALN

We are given a string. Determine the longest palindromic substring without any adjacent duplicates.

For example:  $S = \text{"ABBCBBA"}$ , longest palindromic substring is "ABBCBBBA" but it contains adjacent duplicates, so the required string is "BCB".

EDIT: If there are multiple such strings then print the lexicographical smallest string.

## SOLUSI

Permasalahan ini bertujuan untuk mencari substring palindrome terpanjang dari sebuah string dengan tidak ada huruf yang sama yang saling berdekatan. Jika ada banyak hasil yang mungkin, tampilkan dalam leksikografi terkecil. Misal dari string "BAADBBD C", maka hasilnya adalah "A".

Berbekal literasi dan referensi. Solusi yang dapat dipakai yaitu dengan Dynamic Programming secara bottom up. Berikut adalah ide yang dipakai.

1. Buat sebuah boolean table  $table[n][n]$ , dimana  $n$  adalah panjang dari string. Table ini nantinya akan diisi secara bottom up.
2. Jika substring adalah palindrome maka  $table[i][j] = \text{true}$ . Selain itu  $table[i][j] = \text{false}$ .
3. Untuk menghitung  $table[i][j]$ , cek nilai dari  $table[i+1][j-1]$ , jika nilainya true dan  $str[i]$  sama dengan  $str[j]$  dan  $i \bmod 2$  sama dengan  $j \bmod 2$ , maka  $table[i][j] = \text{true}$ .
4. Untuk mengatasi huruf sama yang berdekatan maka ketika  $str[i]$  sama dengan  $str[i+1]$ , maka  $table[i][j] = \text{false}$
5. Selain diatas maka  $table[i][j] = \text{false}$ .
6. Terminal case dimulai dari  $maxLength = 1$  dengan  $table[i][j] = \text{true}$ .
7. Pada saat  $table[i][j]$  bernilai true, simpan panjang palindrome dan index pertama substring ke dalam table array  $log[n][2]$ . Selain itu jangan simpan.
8. Setelah selesai dengan perhitungan inti, lakukan komparasi untuk mendapatkan substring palindrome dengan length terpanjang. Karena ada kemungkinan substring palindrome dengan length terpanjang tidak hanya 1. Lakukan komparasi tiap substring untuk ditampilkan dalam leksikografi terkecil.

		0	1	2	3	4	5	6
		A	B	B	C	B	B	A
0	A	1	0	0	0	0	0	0
1	B	0	1	0	0	0	0	0
2	B	0	0	1	0	1	0	0
3	C	0	0	0	1	0	0	0
4	B	0	0	0	0	1	0	0
5	B	0	0	0	0	0	1	0
6	A	0	0	0	0	0	0	1

Contoh  $table[n][n]$  dari string "ABBCBBA"

## PSEUDOCODE

SOLVE(A)

```
1.  n = A.length
2.  let table[0..n][0..n] be a new array
3.  let log[0..n][0..2]
4.  maxLength = 1
5.  cnt = n
6.  for i=0 to n-1
7.      table[i][i] = 1
8.      log[i][0] = i
9.      log[i][1] = 1
10. for k=3 to n
11.     for i=0 to n-1
12.         j = i + k - 1
13.         if (table[i + 1][j - 1]) and (str[i] == str[j]) and (i mod 2 == j mod 2)
14.             if str[i]==str[i+1]
15.                 table[i][j] = 0
16.             else
17.                 table[i][j] = 1
18.                 log[cnt][0] = i
19.                 log[cnt][1] = k
20.                 cnt = cnt + 1
21.                 if(k>maxLength)
22.                     maxLength = k
23. flag = 1
24. let temp[0 .. maxLength+1] be a new array
25. let hasil[0 .. maxLength+1] be a new array
26. for i=0 to cnt-1
27.     if log[i][1]==maxLength
28.         l = 0
29.         for j=log[i][0] to log[i][0]+maxLength-1
30.             temp[l] = str[j]
31.         temp[maxLength] = '\0'
32.         if flag or (strcmp(hasil,temp)>0)
33.             strcpy(hasil,temp)
34.             flag = 0
35. return hasil
```

## SOURCECODE

```
#include <stdio.h>
#include <string.h>
int main(){
    int t;
    scanf("%d",&t);
    char str[5001];
    while(t--){
        scanf("%s",str);
        int n = strlen(str);
        char table[n][n];
        memset(table, 0, sizeof(table));
        int log[n][2];
        memset(log, 0, sizeof(log));
        int maxLength = 1;
        for (int i = 0; i < n; ++i){
            table[i][i] = 1;
            log[i][0] = i;
            log[i][1] = 1;
        }
        int cnt=n;
        for (int k = 3; k <= n; ++k) {
            for (int i = 0; i < n - k + 1; ++i) {
                int j = i + k - 1;
                if(table[i + 1][j - 1] && str[i] == str[j] && i%2==j%2){
                    if(str[i]==str[i+1])
                        table[i][j] = 0;
                    else{
                        table[i][j] = 1;
                        log[cnt][0] = i;
                        log[cnt][1] = k;
                        cnt++;
                        if(k>maxLength)
                            maxLength = k;
                    }
                }
            }
        }
        char flag = 1;
        char temp[maxLength+1],hasil[maxLength+1];
        for (int i=0;i<cnt;i++){
            if(log[i][1]==maxLength){
                for (int j=log[i][0],l=0;j<log[i][0]+maxLength;j++,l++){
                    temp[l]=str[j];
                }
                temp[maxLength]='\0';
                if(flag || strcmp(hasil,temp)>0){
                    strcpy(hasil,temp);
                    flag = 0;
                }
            }
        }
        puts(hasil);
    }
    return 0;
}
```

SUBMISSIONS

Fajar: submissions  
Longest palindrome with no adjacent duplicates

ID		DATE	PROBLEM	RESULT	TIME	MEM	LANG
27947657		2021-05-24 15:58:01	Longest palindrome with no adjacent duplicates	<b>accepted</b> <a href="#">edit</a> <a href="#">ideone it</a>	0.00	5.4M	C