

## Zad 2.

niedziela, 7 maja 2023 14:54

2. (1.5pkt) Ułóż algorytm, który dla danego ciągu znajduje długość najdłuższego jego podciągu, który jest palindromem.

```
// find length of longest palindrome subsequence
int longest_palindrome(string s) {
    int n = s.length();
    int** dp = new int*[n];
    for (int i = 0; i < n; i++) dp[i] = new int[n];
    for (int i = 0; i < n; i++) dp[i][i] = 1; // main diagonal
    // fill column by column, from diagonal up
    for (int r = 1; r < n; r++) {
        for (int l = r-1; l >= 0; l--) {
            if (s[l] == s[r]) {
                if (l+1 == r) dp[l][r] = 2; // interval of length 2
                else dp[l][r] = dp[l+1][r-1] + 2;
            }
            else dp[l][r] = max(dp[l+1][r], dp[l][r-1]);
        }
    }
    int result = dp[0][n-1];
    // cleanup
    for (int i = 0; i < n; i++) delete [] dp[i];
    delete [] dp;
    // result
    return result;
}
```

odtworzenie palindromu:

1. zaczynamy w  $dp[0][n-1]$
2. patzimy skąd przyszliśmy:
  - a)  $\nwarrow$  - dokładamy odpowiednie literki do środka  
wyniku (aa  $\rightarrow$  abba „rozpychamy”)
  - $\leftarrow$   $\vee$   $\downarrow$  - nic nie robimy tylko idziemy dalej
3. jak dojdziemy na początek to fajnie