

Zad 1.

niedziela, 7 maja 2023 14:52

1. (1pkt) Ułóż algorytm znajdujący najtańszą drogę przejścia przez tablicę, w którym oprócz ruchów dopuszczalnych w wersji problemu prezentowanej na wykładzie, dozwolone są także ruchy w górę i w dół tablicy.

```
// calculate length of shortest path
int array_walk(unsigned** tab, int h, int w) {
    // initialize dp
    int** dp = new int*[h];
    for (int i = 0; i < h; i++) dp[i] = new int[w];
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            if (j > 0) dp[i][j] = 0;
            else dp[i][j] = tab[i][j];
        }
    }
    // fill dp, column by column
    for (int j = 1; j < w; j++) {
        for (int i = 0; i < h; i++) {
            // update from left-up, left, left-down
            int left_up = dp[max(i-1, 0)][j-1];
            int left = dp[i][j-1];
            int left_down = dp[min(i+1, h-1)][j-1];
            dp[i][j] = min(left_up, left_down);
            dp[i][j] = min(dp[i][j], left);
            // update from up
            int up = dp[max(i-1, 0)][j];
            dp[i][j] = min(dp[i][j], up);
            // update TO up
            dp[max(i-1, 0)][j] = min(dp[max(i-1, 0)][j], dp[i][j]);
        }
    }
    // find min
    int result = INT_MAX;
    for (int i = 0; i < h; i++)
        result = min(result, dp[i][w-1]);
    // cleanup
    for (int i = 0; i < h; i++) delete [] dp[i];
    delete [] dp;
    return result;
}
```

odtworzenie tak jak w przykładzie z wykładu