

# Kolejka jednomonotoniczności

- złożoność obliczeniowa:  $O(2n)/O(1)$
- pozwala na wyznaczenie minimum/maksimum na przedziale w czasie stałym

## IMPLEMENTACJA

```
#include<bits/stdc++.h>

using namespace std;

int n;
int a[1000007]{};

list<pair<int, int>> in(int x, int a, bool b){
    list<pair<int, int>> q;
    if(b){//kolejka minimum
        while(!q.empty() && x <= q.back().first) q.pop_back();
        q.push_back({x, a});
    }
    else{//kolejka maksimum
        while(!q.empty() && x >= q.back().first) q.pop_back();
        q.push_back({x, a});
    }
    return q;
}

int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    list<pair<int, int>> max;
    list<pair<int, int>> min;

    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i];

    int i = 0, j = 0;
    while(j != n - 1){
        //jeśli zwiększamy gąsienicę
        max = in(a[i], i, 0);
        min = in(a[i], i, 1);
        //jeśli skracamy gąsienicę
        while(min.front().second < j) min.pop_front();
        while(max.front().second < j) max.pop_front();
    }
}
```



do minimum i maksimum musimy mieć osobne kolejki