1 Struktury danych 1

2 Matma 1

1 Struktury danych

Drzewo falkowe

Opis: Obsługuje zapytania typu *podaj k-ty najmniejszy na przedziale* itp. na statycznej tablicy. Jeżeli czas albo pamięć są ciasne warto przeskalować liczby. Niszczy tablicę.
Czas: $\mathcal{O}(\log A)$

```
struct node {
  int lo, hi;
  vector<int> s;
  node* l = 0;
  node* r = 0;
  node(int _lo, int _hi, auto st, auto ed) {
    lo = _lo;
    hi = _hi;
    if (lo + 1 < hi) {
      int mid = (lo + hi) / 2;
      s.reserve(ed - st + 1);
      s.push_back(0);
      for (auto it = st; it != ed; it++) {
        s.push_back(s.back() + (*it < mid));
      }
      auto k = stable_partition(
          st, ed, [&](int x) { return x < mid; });
      if (k != st) l = new node(lo, mid, st, k);
      if (k != ed) r = new node(mid, hi, k, ed);
    }
  }
  int kth(int a, int b, int k) {
    if (lo + 1 == hi) return lo;
    int x = s[a];
    int y = s[b];
    return k < y - x ? l->kth(x, y, k)
                     : r->kth(a - x, b - y, k - (y - x));
  }
  int count(int a, int b, int k) {
    if (lo >= k) return 0;
    if (hi <= k) return b - a;
    int x = s[a];
    int y = s[b];
    return (l ? l->count(x, y, k) : 0) +
           (r ? r->count(a - x, b - y, k) : 0);
  }
  int freq(int a, int b, int k) {
    if (k < lo || hi <= k) return 0;
    if (lo + 1 == hi) return b - a;
    int x = s[a];
    int y = s[b];
    return (l ? l->freq(x, y, k) : 0) +
           (r ? r->freq(a - x, b - y, k) : 0);
  }
};
```

Ordered set

Opis: Alternatywnie można użyć treapa albo trie.
Stosowanie: `s.find_by_order(k)` i `s.order_of_key(k)`.
Czas: $\mathcal{O}(\log n)$ z dużą stałą.

```
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template <typename T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
                         tree_order_statistics_node_update>;
```

2 Matma

2.1 Mnożniki Lagrange'a

Jeżeli optymalizujemy $f(x_1, \ldots, x_n)$, pod warunkami typu $g_k(x_1, \ldots, x_n) = 0$ to $x_1, \ldots, x_n$ jest ekstremum lokalnym tylko jeżeli gradient $\nabla f(x_1, \ldots, x_n)$ jest kombinacją liniową gradientów $\nabla g_k(x_1, \ldots x_n)$.