Moscow Institute of Physics and Technology

My Pity

Fedor Alekseev, Dmitry Ivaschenko, Daria Kolodzey

Whatever contest

today

Strings (1)

AhoCorasick.h

Description: on-line tracking of the set of suffixes of a text that are prefixes of some words from a dictionary.

48 lines

```
struct Node {
  Node(int parent, Char ch): parent(parent), lastChar(ch) {
    memset(trans, -1, sizeof trans);
  }
  int trans[Alph];
  int link = 0, nextTerm = -1, termId = -1, parent;
  Char lastChar;
};
struct AhoCorasick {
  void addWord(const string& word, int id) {
    int v = 0;
    for (int ch : word) {
      ch -= 'a';
      auto& u = n[v].trans[ch];
      if (u == -1) {
        n.emplace_back(v, ch);
        u = int(n.size() - 1);
      }
      v = u;
    }
    n[v].termId = id;
  }
  void build() {
    queue<int> q;
    for (auto& tr : n[0].trans) {
      if (tr != -1) {
        q.push(tr);
      } else
        tr = 0;
    }
    while (!q.empty()) {
      auto v = q.front();
      q.pop();
      auto& li = n[v].link;
      auto par = n[v].parent;
      li = (par ? n[n[par].link].trans[n[v].lastChar] : 0);
      n[v].nextTerm = n[li].termId != -1 ? li : n[li].nextTerm;
      for (Char ch = 0; ch < Alph; ++ch) {
        if (auto& u = n[v].trans[ch]; u != -1) {
          q.push(u);
        } else
          u = n[li].trans[ch];
      }
    }
  }
private:
  vector<Node> n{{-1, 0}};
};
```

PrefixFunction.h

Description: pi[x] is the length of the longest prefix of s that ends at x, other than s[0..x] itself

10 lines

```
vector<size_t> pi(const string& s) {
  vector<size_t> p(s.size(), 0);
  for (size_t i = 1; i < s.size(); ++i) {
    auto px = p[i - 1];
    while (px && s[i] != s[px])
      px = p[px - 1];
    p[i] = px + (s[i] == s[g]);
  }
  return p;
}
```

ZFunction.h

Description: z[x] is max L: s[x:x+L] == s[:L]

11 lines

```
vector<size_t> zFun(const string& s) {
  vector<size_t> z(s.size(), 0);
  for (size_t left = 0, right = 0, i = 1; i < s.size(); ++i) {
    z[i] = (i < right ? min(right - i, z[i - left]) : 0);
    while (i + z[i] < s.size() && s[i + z[i]] == s[z[i]])
      ++z[i];
    if (i + z[i] > right)
      tie(left, right) = {i, i + z[i]};
  }
  return z;
}
```