

ACM-ICPC TEAM REFERENCE DOCUMENT

HSE-NN 2

Содержание

1	Шаблон	2
2	Алгоритмы на строки	2
2.1	Префикс-функция	2
2.2	Z-функция	2
3	Алгоритмы на графах	2
3.1	Алгоритм Дейкстры $O(n^2)$	2
3.2	Алгоритм Дейкстры $O(\log(n) \cdot m)$. .	2

1 Шаблон

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <set>
#include <queue>
#include <utility>
#include <iomanip>
#include <cstdio>
#include <cstdlib>
#include <numeric>
#include <cmath>
#include <stack>
#include <map>
#include <deque>
#include <sstream>
using namespace std;
#define int long long
typedef vector<int> vi;
typedef vector<pair<int, int>> vii;
typedef long long ll;
typedef long double ld;
// #define pi M_PI
#define all(x) (x).begin(), (x).end()
#define pb push_back
#define re return
#define fr(x) for(int i = 0; i < (x); i++)
const int inf = 1000000000 + 7;
signed main() {
}
```

2 Алгоритмы на строки

2.1 Префикс-функция

```
vector<int> prefix_function (string s) {
    int n = (int) s.length();
    vector<int> pi (n);
    for (int i=1; i<n; ++i) {
        int j = pi[i-1];
        while (j > 0 && s[i] != s[j])
            j = pi[j-1];
        if (s[i] == s[j]) ++j;
        pi[i] = j;
    }
    return pi;
}
```

2.2 Z-функция

```
vector<int> z_function (string s) {
    int n = (int) s.length();
    vector<int> z (n);
    for (int i=1, l=0, r=0; i<n; ++i) {
        if (i <= r)
            z[i] = min (r-i+1, z[i-l]);
        while (i+z[i] < n && s[z[i]] == s[i+z[i]])
            ++z[i];
        if (i+z[i]-1 > r)
            l = i, r = i+z[i]-1;
    }
    return z;
}
```

3 Алгоритмы на графах

3.1 Алгоритм Дейкстры $O(n^2)$

was - брали вершину или нет

v - список смежности

d - массив расстояний для точки x

```
int d[2001];
int was[2001];
vector<pair<int, int>> v[2001];
int n;
void dijkstra(int x) {
    for (int i = 0; i < n; i++)
        d[i] = inf;
    d[x] = 0;
    for (int it = 0; it < n; it++)
    {
        int id = -1;
        for (int i = 0; i < n; i++)
            if (!was[i] && (id == -1 || d[id] > d[i]))
                id = i;
        was[id] = 1;
        for (auto p : v[id]) {
            int y = p.first;
            int t = p.second;
            d[y] = min(d[y], d[id] + t);
        }
    }
}
```

3.2 Алгоритм Дейкстры $O(\log(n) \cdot m)$

d - массив расстояний для точки x

```
int d[3001];
vector<pair<int, int>> v[3001];
bool f(int x, int y) {
    if (d[x] != d[y])
        return d[x] < d[y];
    return x < y;
}
set <int, bool*(int, int)> s(f);
void dijkstra(int x) {
    x--;
    for (int i = 0; i <= n; i++)
    {
        d[i] = inf;
    }
    d[x] = 0;
    s.insert(x);
    while (!s.empty()) {
        int x = *s.begin();
        s.erase(x);
        for (auto p : v[x]) {
            int y = p.first;
            int t = p.second;
            if (d[y] > d[x] + t) {
                s.erase(y);
                d[y] = d[x] + t;
                s.insert(y);
            }
        }
    }
}
```