Algorithm Library

magic::team.getname()

South China Normal University

August 29, 2024

Contents

一切的开始

宏定义

- 需要 C++11

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  using LL = long long;
4  #define FOR(i, x, y) for (decay<decltype(y)>::type i = (x), _##i = (y); i < _##i; ++i)
5  #define FORD(i, x, y) for (decay<decltype(x)>::type i = (x), _##i = (y); i > _##i; --i)
6  #ifdef zerol
7  #define dbg(x...) do { cout << "\033[32;1m" << #x << " -> "; err(x); } while (0)
8  void err() { cout << "\033[39;0m" << endl; }
9  template<template<typename...> class T, typename t, typename... A>
10 void err(T<t> a, A... x) { for (auto v: a) cout << v << ' '; err(x...); }
11 template<typename T, typename... A>
12 void err(T a, A... x) { cout << a << ' '; err(x...); }
13 #else
14 #define dbg(...)
15 #endif
16 // ------------------------------------------------------------------------
```

图论

LCA

- 倍增

```
1  void dfs(int u, int fa) {
2      pa[u][0] = fa; dep[u] = dep[fa] + 1;
3      FOR (i, 1, SP) pa[u][i] = pa[pa[u][i - 1]][i - 1];
4      for (int& v: G[u]) {
5          if (v == fa) continue;
6          dfs(v, u);
7      }
8  }
9
10 int lca(int u, int v) {
11     if (dep[u] < dep[v]) swap(u, v);
12     int t = dep[u] - dep[v];
13     FOR (i, 0, SP) if (t & (1 << i)) u = pa[u][i];
14     FORD (i, SP - 1, -1) {
15         int uu = pa[u][i], vv = pa[v][i];
16         if (uu != vv) { u = uu; v = vv; }
17     }
18     return u == v ? u : pa[u][0];
19 }
```

计算几何

二维几何：点与向量

```
1  #define y1 yy1
2  #define nxt(i) ((i + 1) % s.size())
3  typedef double LD;
4  const LD PI = 3.14159265358979323846;
5  const LD eps = 1E-10;
6  int sgn(LD x) { return fabs(x) < eps ? 0 : (x > 0 ? 1 : -1); }
7  struct L;
8  struct P;
9  typedef P V;
10 struct P {
11     LD x, y;
12     explicit P(LD x = 0, LD y = 0): x(x), y(y) {}
13     explicit P(const L& l);
14 };
15 struct L {
```
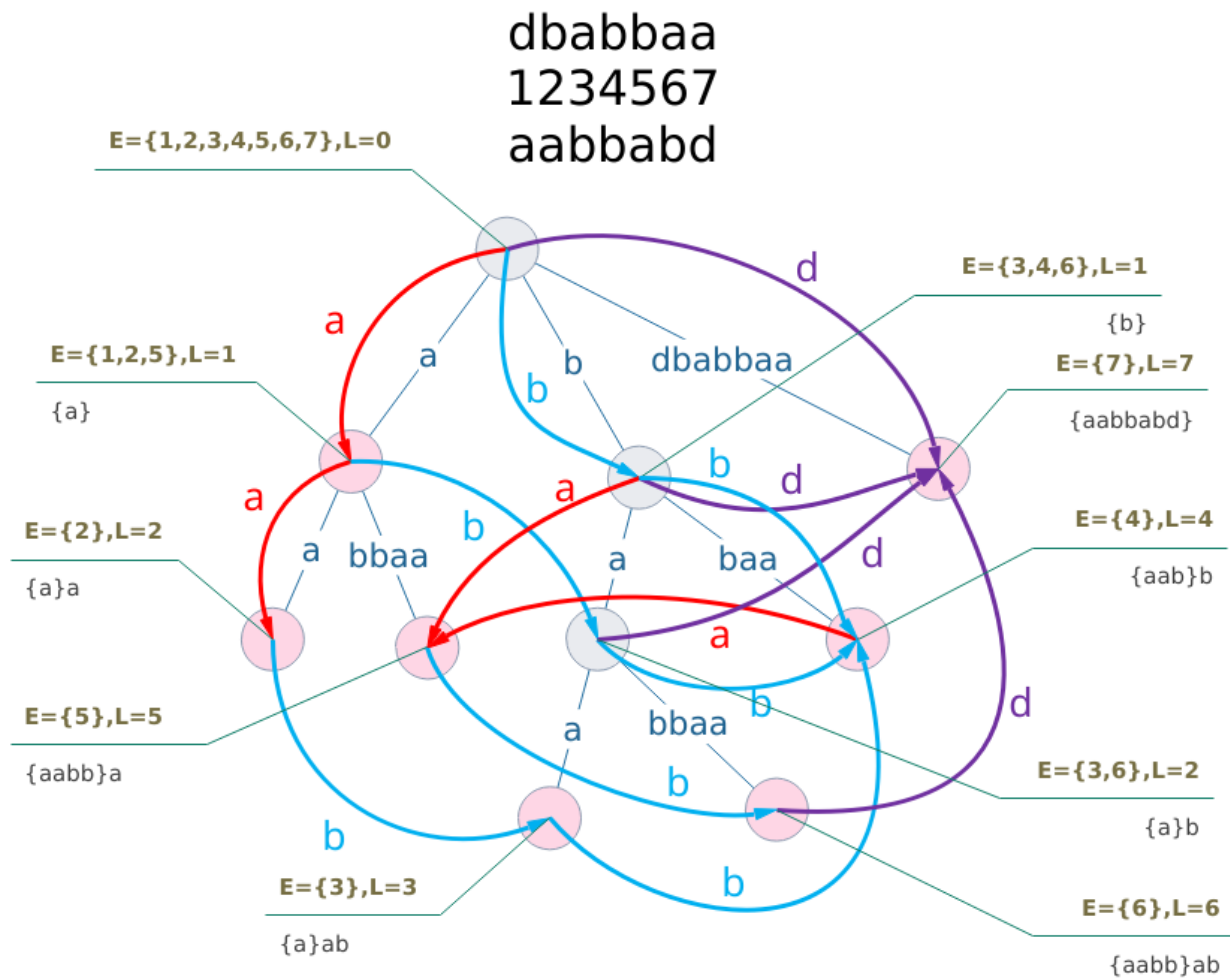
```cpp
        P s, t;
        L() {}
        L(P s, P t): s(s), t(t) {}
};

P operator + (const P& a, const P& b) { return P(a.x + b.x, a.y + b.y); }
P operator - (const P& a, const P& b) { return P(a.x - b.x, a.y - b.y); }
P operator * (const P& a, LD k) { return P(a.x * k, a.y * k); }
P operator / (const P& a, LD k) { return P(a.x / k, a.y / k); }
inline bool operator < (const P& a, const P& b) {
    return sgn(a.x - b.x) < 0 || (sgn(a.x - b.x) == 0 && sgn(a.y - b.y) < 0);
}
bool operator == (const P& a, const P& b) { return !sgn(a.x - b.x) && !sgn(a.y - b.y); }
P::P(const L& l) { *this = l.t - l.s; }
ostream &operator << (ostream &os, const P &p) {
    return (os << "(" << p.x << "," << p.y << ")");
}
istream &operator >> (istream &is, P &p) {
    return (is >> p.x >> p.y);
}

LD dist(const P& p) { return sqrt(p.x * p.x + p.y * p.y); }
LD dot(const V& a, const V& b) { return a.x * b.x + a.y * b.y; }
LD det(const V& a, const V& b) { return a.x * b.y - a.y * b.x; }
LD cross(const P& s, const P& t, const P& o = P()) { return det(s - o, t - o); }
// ----------------------------------------
```

字符串

后缀自动机


```
dbabbaa
1234567
aabbabd
```

E={1,2,3,4,5,6,7},L=0

E={3,4,6},L=1
{b}

E={7},L=7
{aabbabd}

E={1,2,5},L=1
{a}

E={2},L=2
{a}a

E={4},L=4
{aab}b

E={5},L=5
{aabb}a

E={3,6},L=2
{a}b

E={3},L=3
{a}ab

E={6},L=6
{aabb}ab

杂项

STL

- copy

```
template <class InputIterator, class OutputIterator>
  OutputIterator copy (InputIterator first, InputIterator last, OutputIterator result);
```