



---

# Stratosphere's XCPC Templates

---

南京大学

平流层 Stratosphere

October 1, 2024

## Contents

<b>0 Header 与约定</b>	<b>1</b>
<b>1 图论</b>	<b>2</b>
1.1 欧拉回路 . . . . .	2
1.2 Tarjan-SCC . . . . .	3
1.3 点双 . . . . .	3
1.4 边双 . . . . .	3
<b>2 数论</b>	<b>5</b>
<b>3 数学</b>	<b>6</b>
<b>4 字符串</b>	<b>7</b>
<b>5 数据结构</b>	<b>8</b>
<b>6 计算几何</b>	<b>9</b>
<b>7 三维计算几何</b>	<b>10</b>
<b>8 杂项</b>	<b>11</b>

## 0 Header 与约定

# 1 图论

## 1.1 欧拉回路

```

1 namespace Euler {
2     bool directed;
3     vector<pii> G[maxn];
4     vector<int> ans;
5     int vis[maxn];
6     int dfs(int x) {
7         vector<int> t;
8         while (G[x].size()) {
9             auto [to, id] = G[x].back();
10            G[x].pop_back();
11            if (!vis[abs(id)]) {
12                vis[abs(id)] = 1, t.push_back(dfs(to)), ans.push_back(id);
13            }
14        }
15        for (int i = 1; i < t.size(); i++) {
16            if (t[i] != x) ans.clear();
17        }
18        return t.size() ? t[0] : x;
19    }
20    int n, m;
21    pii e[maxn];
22    int deg[maxn], vv[maxn];
23    void clr() {
24        for (int i = 1; i <= n; i++) G[i].clear(), deg[i] = vv[i] = 0;
25        for (int i = 1; i <= m; i++) vis[i] = 0;
26        ans.clear();
27        n = m = 0;
28    }
29    void addedge(int x, int y) {
30        chkmax(n, x), chkmax(n, y);
31        ++m;
32        e[m] = {x, y};
33        if (directed) {
34            G[x].push_back({y, m});
35            ++deg[x], --deg[y], vv[x] = vv[y] = 1;
36        } else {
37            G[x].push_back({y, m});
38            G[y].push_back({x, -m});
39            ++deg[x], ++deg[y], vv[x] = vv[y] = 1;
40        }
41    }
42    using vi = vector<int>;
43    pair<vi, vi> work() {
44        if (!m) return clr(), pair<vi, vi>{{1}, {}};
45        int S = 1;
46        for (int i = 1; i <= n; i++)
47            if (vv[i]) S = i;
48        for (int i = 1; i <= n; i++)
49            if (deg[i] > 0 && deg[i] % 2 == 1) S = i;
50        dfs(S);
51        if ((int)ans.size() != m) return clr(), pair<vi, vi>();
52        reverse(ans.begin(), ans.end());
53        vi ver, edge = ans;
54        if (directed) {
55            ver = {e[ans[0]].fir};

```

```

56     for (auto t : ans) ver.push_back(e[t].sec);
57   } else {
58     ver = {ans[0] > 0 ? e[ans[0]].fir : e[-ans[0]].sec};
59     for (auto t : ans) ver.push_back(t > 0 ? e[t].sec : e[-t].fir);
60   }
61 }
62 } // namespace Euler

```

## 1.2 Tarjan-SCC

```

1 void tarjan(int u) {
2   dfn[u] = low[u] = ++tim;
3   in[u] = 1;
4   st[++top] = u;
5   for (int v : G[u]) {
6     if (!dfn[v])
7       tarjan(v), ckmin(low[u], low[v]);
8     else if (in[v])
9       ckmin(low[u], dfn[v]);
10  }
11  if (dfn[u] == low[u]) {
12    ++totc;
13    int x;
14    do { x = st[top--], in[x] = 0, bel[x] = totc; } while (x != u);
15  }
16 }

```

## 1.3 点双

```

1 int T; // assign = n
2 void tarjan(int u, int fa) {
3   dfn[u] = low[u] = ++tim;
4   stk[++top] = u;
5   for (int v : G[u]) {
6     if (v == fa) continue;
7     if (!dfn[v])
8       dfs(v, u), ckmin(low[u], low[v]);
9     else
10      ckmin(low[u], dfn[v]);
11  }
12  if (fa && low[u] >= dfn[fa]) {
13    int y;
14    ++T;
15    do {
16      y = stk[top--];
17      G2[T].push_back(y), G2[y].push_back(T);
18    } while (y != u);
19    G2[T].push_back(fa), G2[fa].push_back(T);
20  }
21 }

```

## 1.4 边双

```

1 // etot should be initialized to 1 !!!
2 void tarjan(int u, int f) {

```

```
3   dfn[u] = low[u] = ++tim;
4   for (int i = head[u]; i; i = e[i].nxt) {
5       int v = e[i].to;
6       if (v == f) continue;
7       if (!dfn[v]) {
8           tarjan(v, u);
9           ckmin(low[u], low[v]);
10          if (low[v] > dfn[u]) vis[i] = vis[i ^ 1] = 1;
11      } else
12          ckmin(low[u], dfn[v]);
13  }
14 }
15 void fill(int u) {
16     col[u] = cnt;
17     for (int i = head[u]; i; i = e[i].nxt) {
18         int v = e[i].to;
19         if (col[v] || vis[i]) continue;
20         fill(v);
21     }
22 }
```

## 2 数论

### 3 数学



## 4 字符串

## 5 数据结构

## 6 计算几何

## 7 三维计算几何

## 8 杂项