

Spanning Tree

```
struct STStep {
    bool reserve(int i) {
        u = F.find(E[i].u);
        v = F.find(E[i].v);
        if (u == v) return 0;
        if (u > v) swap(u,v);
        R[v].reserve(i); return 1;}

    bool commit(int i) {
        if (R[v].check(i)) { F.link(v, u); return 1;}
        else return 0;  };}

void ST(res* R, edge* E, int m, int n, int psize) {
    disjointSet F(n);
    speculative_for(STStep(E, F, R), 0, m, psize);}
```