**Алгоритм Крускала**

**Алгоритм Крускала** (или **алгоритм Краскала**) — эффективный [алгоритм](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) построения [минимального остовного дерева](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BE%D1%81%D1%82%D0%BE%D0%B2%D0%BD%D0%BE%D0%B5_%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE) взвешенного связного [неориентированного графа](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84).

//n-количество вершин, m-количество ребер

struct edge

{

int a,b, cost;

};

(pair < int, pair < int, int > > e;)

typedef pair < int, pair < int, int > > PIP;

PIP e[M];

vector <PIP> ans;

#define ff first;

#define ss second;

void kruskal ()

{

sort(e, e+m);

for (int i=0; i<m; i++)

{

int a = find\_set(e[i].ss.ss);

int b = find.set(e[i].ss.ss);

if (a!=b)

{

union\_sets(a,b);

ans.push\_back(e[i]);

if (ans.size()==n-1) {break;}

}

}

}

**СНМ**

Система непересекающихся множеств (англ. disjoint-set, или union–find data structure) — структура данных, которая позволяет администрировать множество элементов, разбитое нанепересекающиеся подмножества.

**Функции.**

void make\_set(int a)

{

p[a]=a;

rank[a]=1;

}

void find\_set(int a)

{

if (a==p[a]) {return a;}

else

{

int res = find\_set(p[a]);

p[a]=res;

return res;

}

}

void union\_sets(int a, int b)

{

a=find\_set(a);

b=find\_set(b);

if (rank[a]>rank[b]) {swap(a,b);}

p[a]=b;

if (rank[a]==rank[b]) {rank[b]++;}

}