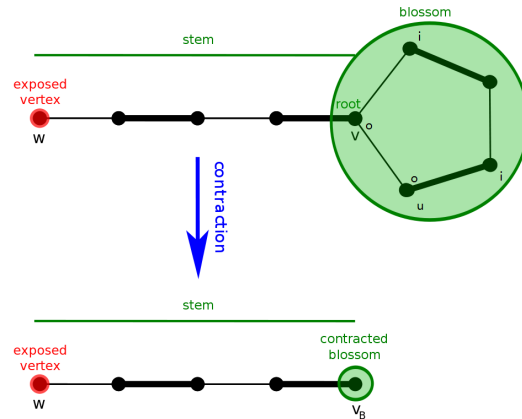


一般图最大匹配

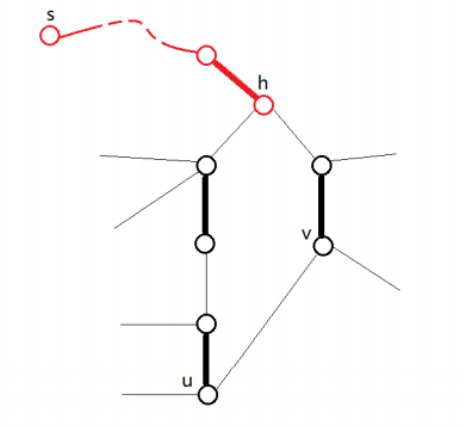
带花树算法 Blossom Algorithm

Idea: 当一张图带有奇环时, 该图不是二分图, 匈牙利算法在遇到奇环时将失效。可以证明, 将奇环缩为一个点 (缩花) 后能保证正确性, 所以我们可以同匈牙利算法一样找增广路, 但是遇到奇环就缩点。



实现上, 每次从未匹配点搜索, 尝试对原图进行二染色 (黑色开始)。考虑 bfs 树, 如果当前点 u 与下一个点 v 已经被缩为一个点 (并查集维护) 或者形成了偶环, 则不管; 如果 v 没有被染色, 就染上白色, 若它已经与 x 匹配好了, 就将 x 染为黑, 且 v 和 x 一并加入树中 (也即将 x 加入队列), 否则找到了一条增广路径, 一路返回更改匹配; 若 v 已经被染色, 形成了奇环, 则将奇环通过开花操作缩掉, 即暴力向上找 $lca(u, v)$ 并一路合并并查集, 同时更改匹配关系使得 $lca(u, v)$ 是那个没有被匹配的点。

注意到, $pre[x]$ 是白点 x 在 bfs 树上的上一个黑点, 而 $match[x]$ 是黑点 x 在 bfs 树上的上一个白点, 理解这一点并画出 bfs 树就不难看懂代码了。



Complexity: $O(VE^2)$

Code:

```
1  int fa[N];
2  int findfa(int x){ return x == fa[x] ? x : fa[x] = findfa(fa[x]); }
3
4  int match[N];
5  int vis[N], pre[N], dfn[N], cnt;
6  queue<int> q;
7  int lca(int x, int y){
8      for(++cnt, x = findfa(x), y = findfa(y); dfn[x] != cnt;){
9          dfn[x] = cnt;
10         x = findfa(pre[match[x]]);
11         if(y) swap(x, y);
12     }
13     return x;
14 }
15 void blossom(int x, int y, int w){
16     while(findfa(x) != w){
17         pre[x] = y, y = match[x];
18         if(vis[y] == 2) vis[y] = 1, q.push(y);
```

```

19         if(findfa(x) == x) fa[x] = w;
20         if(findfa(y) == y) fa[y] = w;
21         x = pre[y];
22     }
23 }
24 bool bfs(int s){
25     for(int i = 1; i <= n; i++) vis[i] = pre[i] = 0, fa[i] = i;
26     while(!q.empty()) q.pop();
27     q.push(s); vis[s] = 1;
28     while(!q.empty()){
29         int u = q.front(); q.pop();
30         for(int i = head[u]; i; i = edge[i].nxt){
31             int v = edge[i].to;
32             if(findfa(u) == findfa(v) || vis[v] == 2) continue;
33             if(!vis[v]){
34                 vis[v] = 2, pre[v] = u;
35                 if(!match[v]){
36                     for(int x = v, lst; x; x = lst)
37                         lst = match[pre[x]], match[x] = pre[x], match[pre[x]] = x;
38                     return true;
39                 }
40                 vis[match[v]] = 1;
41                 q.push(match[v]);
42             }
43             else{
44                 int l = lca(u, v);
45                 blossom(u, v, l);
46                 blossom(v, u, l);
47             }
48         }
49     }
50     return false;
51 }
52
53 int main(){
54     scanf("%d%d", &n, &m);
55     for(int i = 1; i <= m; i++){
56         int u, v; scanf("%d%d", &u, &v);
57         addEdge(u, v), addEdge(v, u);
58     }
59     for(int i = 1; i <= n; i++)
60         if(!match[i])
61             ans += bfs(i);
62     printf("%d\n", ans);
63     for(int i = 1; i <= n; i++)
64         printf("%d ", match[i]);
65     return 0;
66 }

```