# 07A2 双连通分量分解：算法 | 实例

#数据结构邓神

## Graph::BCC()

```
#define hca(x) ( fTime(x) ) //利用此处闲置的fTime
template <typename Tv, typename Te>
void Graph<Tv, Te>::BCC( int v, int & clock, Stack<int> & S ) {
   hca(v) = dTime(v) = ++clock; status(v) = DISCOVERED; S.push(v);
   for ( int u = firstNbr(v); -1 < u; u = nextNbr(v, u) )
      switch ( status(u) )
         { /* ... 视u的状态分别处理 ... */ }
   status(v) = VISITED; //对v的访问结束
}
#undef hca
```

## switch ( status(u) )

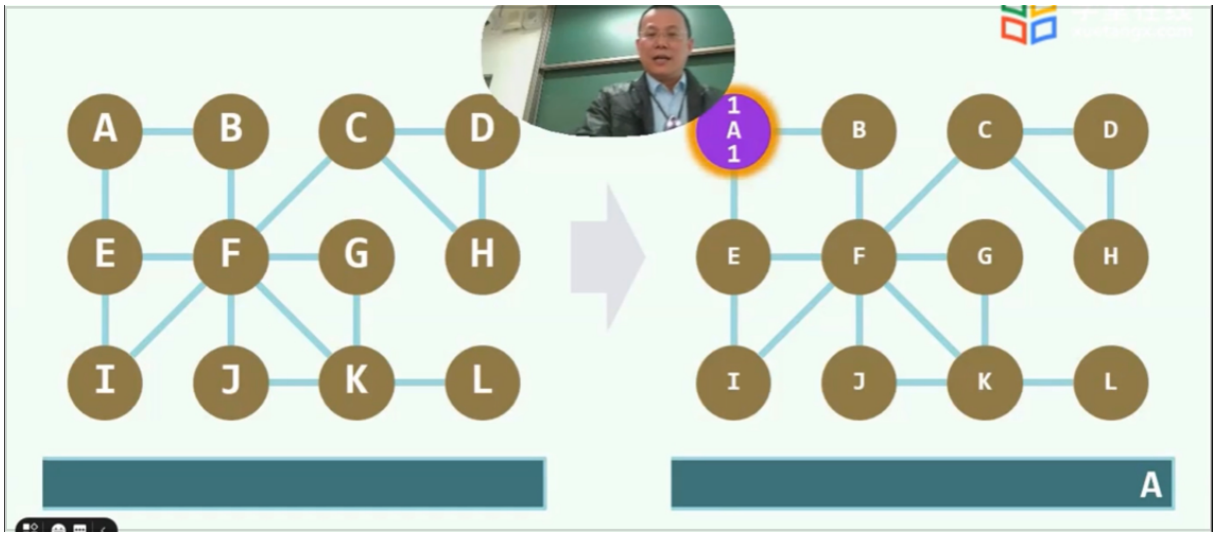
```
case UNDISCOVERED:
   parent(u) = v; type(v, u) = TREE; //拓展树边
   BCC( u, clock, S ); //从u开始遍历，返回后...
   if ( hca(u) < dTime(v) ) //若u经后向边指向v的真祖先
      hca(v) = min( hca(v), hca(u) ); //则v亦必如此
   else //否则，以v为关节点（u以下即是一个BCC，且其中顶点此时正集中于栈S的顶部）
      while ( u != S.pop() ); //弹出当前BCC中（除v外）的所有节点
                              //可视需要做进一步处理
   break;
```


```
#define hca(x) (fTime(x))
template <typename Tv,typename Te>
void Graph<Tv,Te>::BCC(int v,int & clock,stack<int> & S){
    hca(v) = dTime(v) = ++clock;
    status(v) = DISCOVERED;
    S.push(v);
    for (int u = firstNbr(v); -1 < u; u = nextNbr(v)) {
        switch(status(u)) {
            case UNDISCOVERD:{
                parent(u) = v;
                type(v,u) = TREE;
                BCC(u,clock,S);
                  // backTrack回来了 判断一下 dTime 和回传hca值
                if (hca(u) < dTime(v)){
                    hca(v) = min(hca(u),hca(v)); // 他的孩子能到这么高，他自己也能够到了更高
                }else { // 没有爸爸节点高
                    while (u != S.top()) { // 找到了连通域和关键节点
                        S.pop(u);

                    }
                }
                break;
            }
            case DISCOVERED:{
```

```
                type(v,u) = BACKWARD;
                if (u != parent(v)){ // 不能是他的父亲
                    hca(v) = min(hca(v),dTime(u));
                }
                break;
            }
            default:{
                type(v,u) = dTime(v) < dTime(u) ? FORWARD : CROSS;
                break;
            }
        }
    }
    status(v) = VISITED;
}
#undef hca
```

## 实例




下面是dTime 上面是hca