Taller ProgComp: Track Grafos

Puntos y Puentes de Articulación

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

September 8, 2024

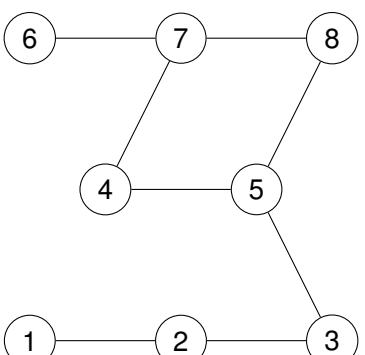
Part I

Puntos y Puentes de Articulación

Puntos de Articulación

Definición

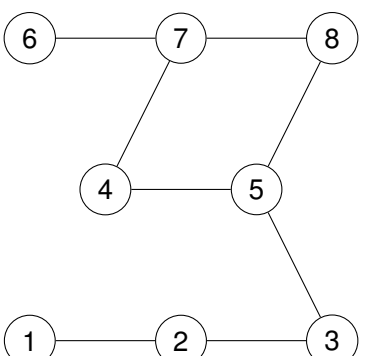
Un punto de articulación es un nodo que al ser borrado separa la componente conexa, en dos componentes conexas.



Puentes de Articulación

Definición

Un puente de articulación es una arista que al ser borrada separa la componente conexa, en dos componentes conexas.



¿Cómo determinarlos?

Solución Inicial

1. Por cada nodo, eliminarlo junto a sus caminos correspondientes.
2. Ejecutar DFS (o BFS) y ver si el número de componentes conexas aumentó.
3. Si la cantidad efectivamente aumentó, entonces la ciudad es un punto de articulación!

¿Cómo determinarlos?

Solución Inicial

1. Por cada nodo, eliminarlo junto a sus caminos correspondientes.
2. Ejecutar DFS (o BFS) y ver si el número de componentes conexas aumentó.
3. Si la cantidad efectivamente aumentó, entonces la ciudad es un punto de articulación!

¿Cuál es el problema?

¿Cómo determinarlos?

Solución Inicial

1. Por cada nodo, eliminarlo junto a sus caminos correspondientes.
2. Ejecutar DFS (o BFS) y ver si el número de componentes conexas aumentó.
3. Si la cantidad efectivamente aumentó, entonces la ciudad es un punto de articulación!

¿Cuál es el problema?
La complejidad es $O(V \cdot (V + E))$. (algo parecido para puentos de articulación)

Modificar DFS

Problema del DFS actual

DFS determinar si un nodo fue visitado o no.

Modificar DFS

Problema del DFS actual

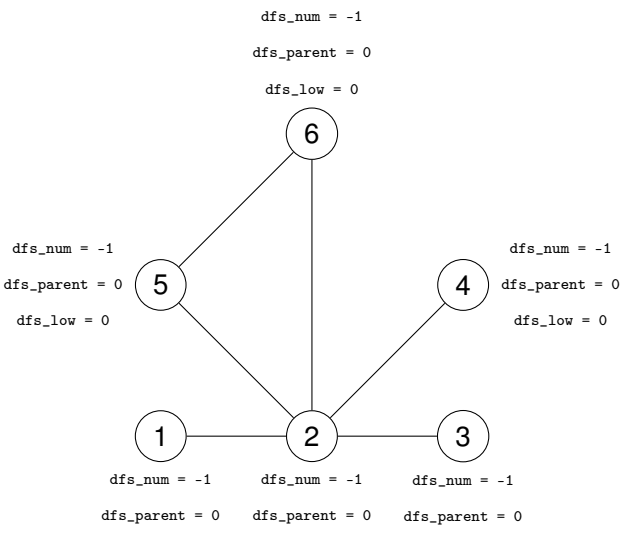
DFS determinar si un nodo fue visitado o no.

Extender DFS

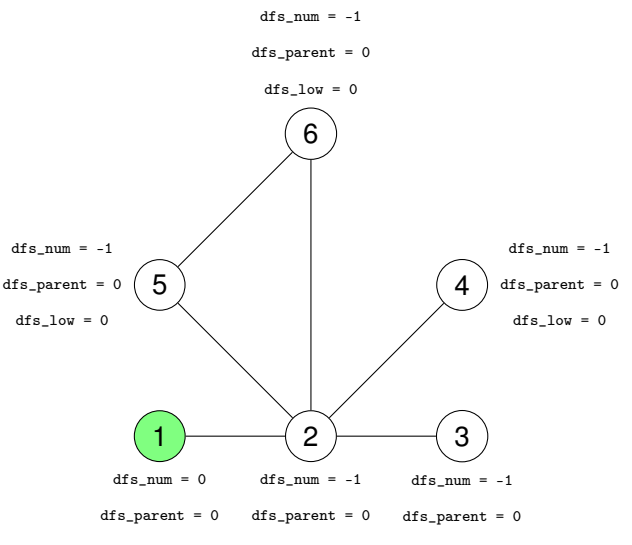
Se sabrá:

- **CUANDO** se llegó al nodo (`dfs_num`)
- **DE DONDÉ** se llegó al nodo (`dfs_parent`)
- **MENOR** `dfs_num` que un vecino del nodo tiene (`dfs_low`)

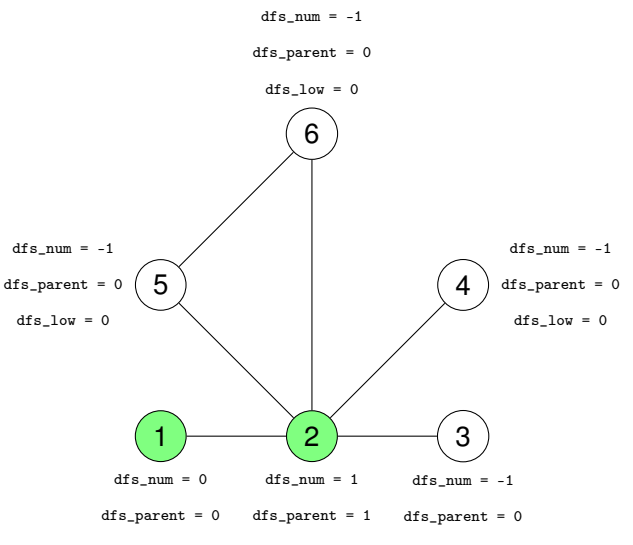
DFS MODIFICADO



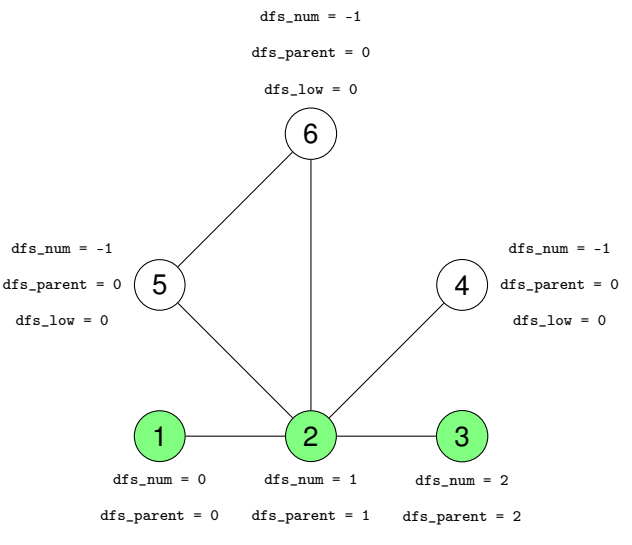
DFS MODIFICADO



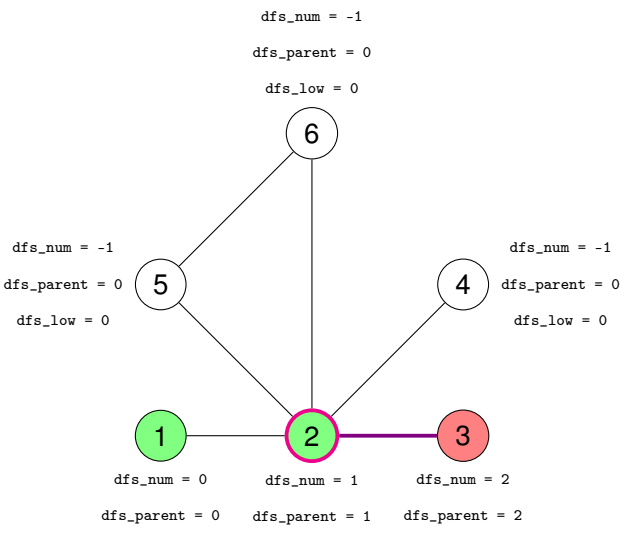
DFS MODIFICADO



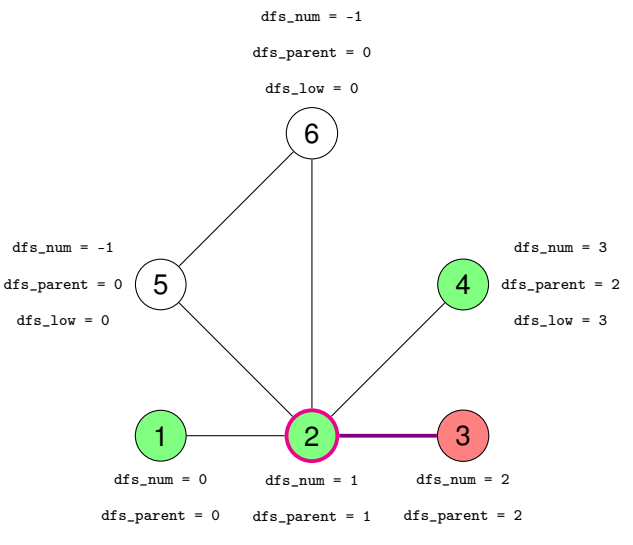
DFS MODIFICADO



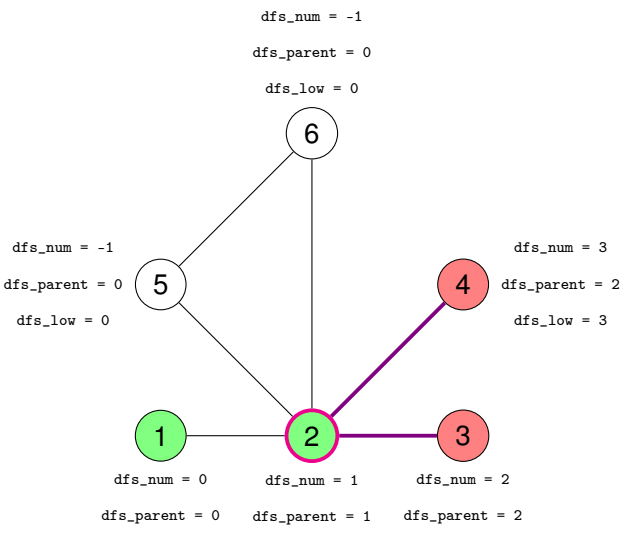
DFS MODIFICADO



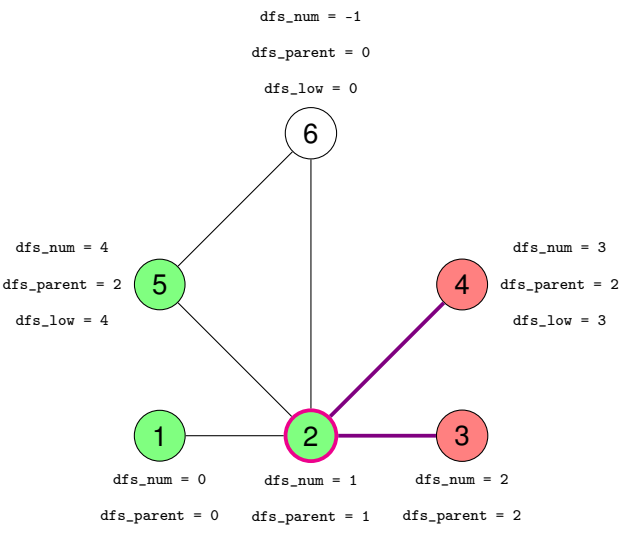
DFS MODIFICADO



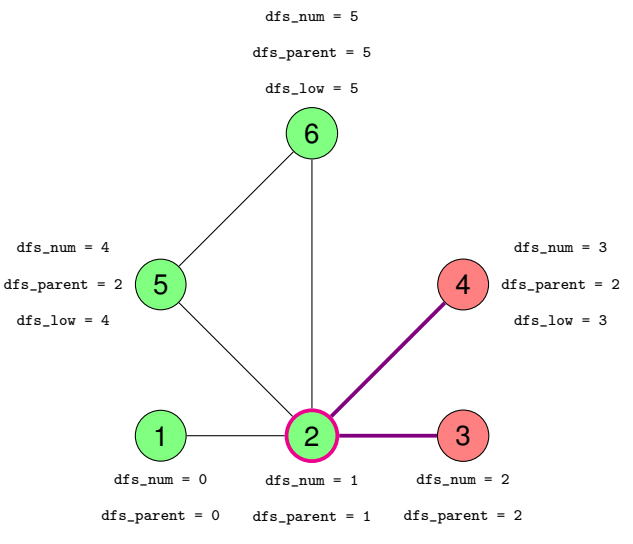
DFS MODIFICADO



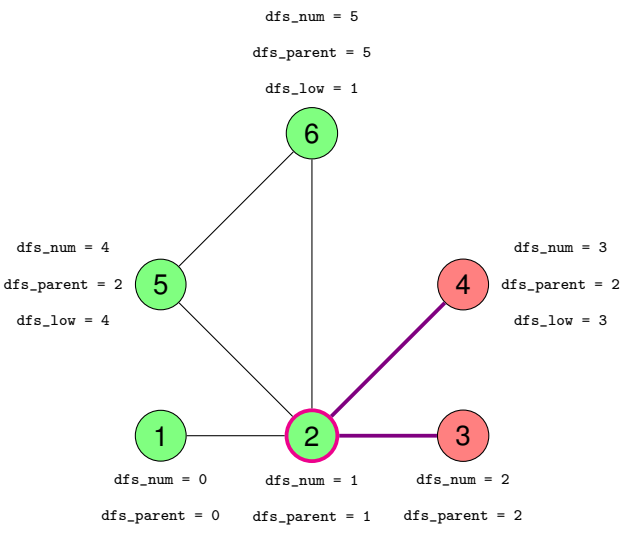
DFS MODIFICADO



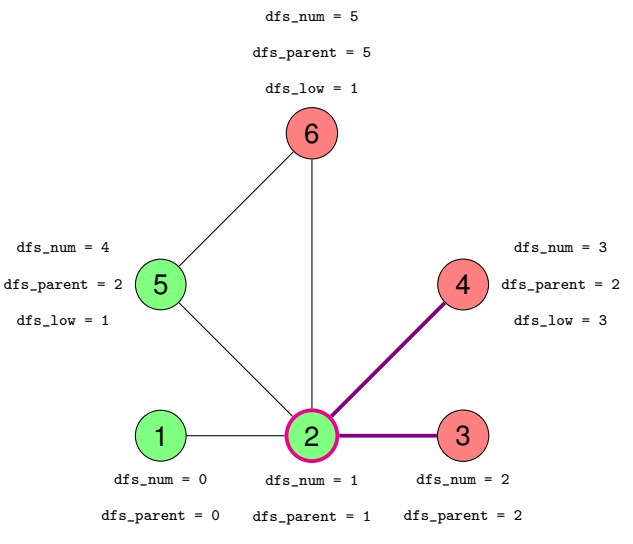
DFS MODIFICADO



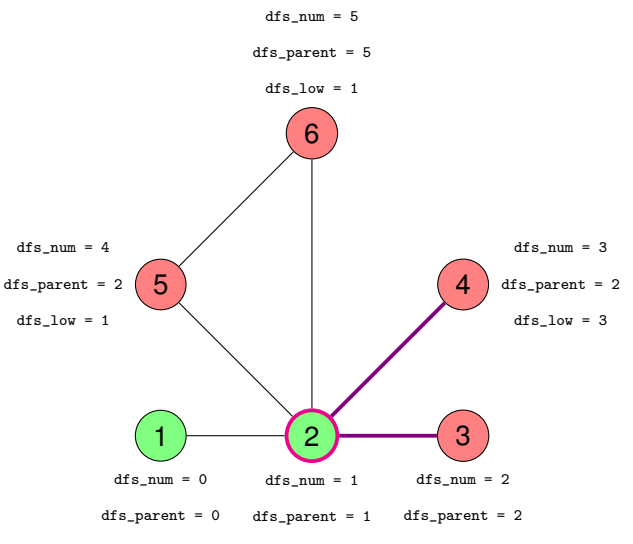
DFS MODIFICADO



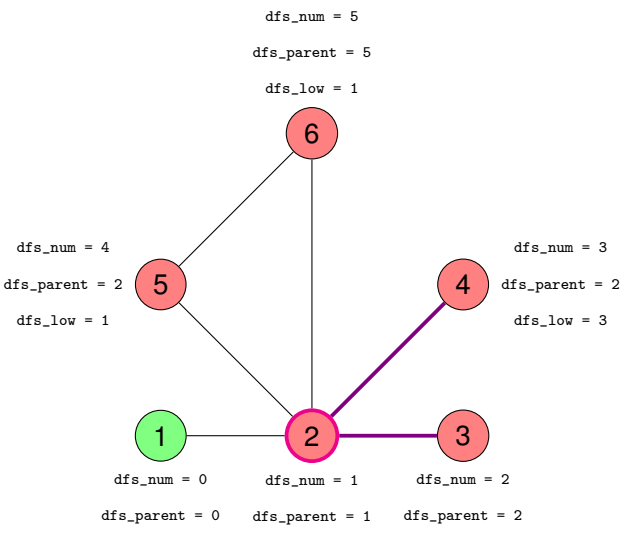
DFS MODIFICADO



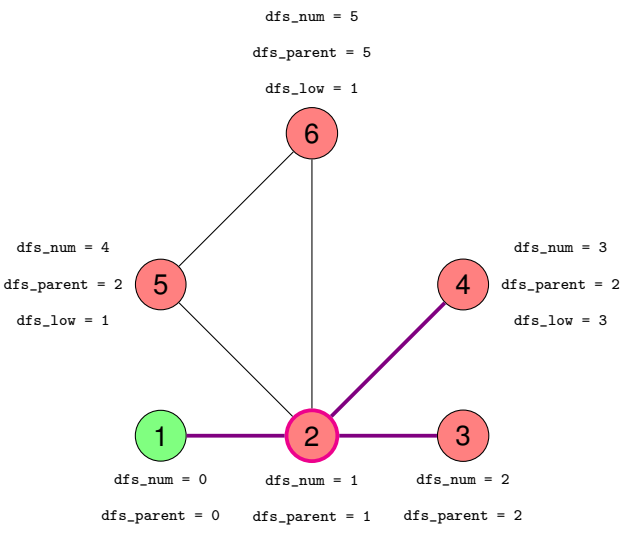
DFS MODIFICADO



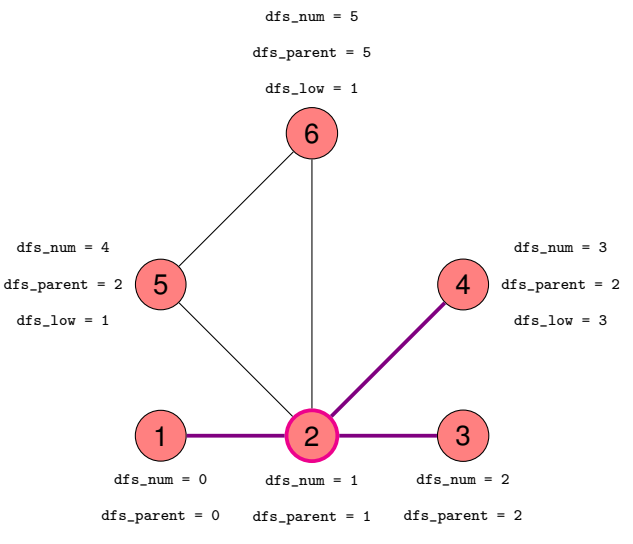
DFS MODIFICADO



DFS MODIFICADO



DFS MODIFICADO



Código DFS modificado

```cpp
int dfs_counter; // cuantas veces se ha llamado a DFS
int dfs_root; // donde comenzamos el DFS en la componente conexa
int root_childen; // cantidad de hijos del nodo que inicia el DFS
vector< int > dfs_num;
vector< int > dfs_low;
vector< int > dfs_parent;
vector< bool > APs; // en la pos i, true si el nodo i es punto de articulacion
vector< pair< int, int > > ABs; // pares {u, v} seran puente de articulacion

void dfs_modificado(int u){
  dfs_low[u] = dfs_num[u] = dfs_counter;
  dfs_counter++;
    for(int i = 0; i < lista[u].size(); i++){
    int v = lista[u][i];
    if(dfs_num[v] == -1) {
        // el padre de v es u
        dfs_parent[v] = u;
        if(u == dfs_root) root_children++;
        dfs_modificado(v);

        if(u != dfs_root && dfs_low[v] >= dfs_num[u]) APs[u] = true;

        if(dfs_low[v] > dfs_num[u]) ABs.push_back(pair< int, int >(u, v));

        dfs_low[u] = min(dfs_low[u], dfs_low[v]);
    } else if(v != dfs_parent[u])
        dfs_low[u] = min(dfs_low[u], dfs_num[v]);
  }
}
```

Codigo para llamar usar el DFS modificado

```cpp
int main(){
  // se asume que existe el codigo para inicializar el grafo correctamente

  dfs_counter = 0;
  dfs_num.resize(n, -1);
  dfs_low.resize(n, 0);
  dfs_parent.resize(n, 0);
  APs.resize(n, false);

  for(int i = 0; i < n; i++){
    if(dfs_num[i] == -1){
      dfs_root = i;
      root_children = 0;
      dfs_modificado(i);
      if(root_children > 1) APs[dfs_root] = true; // caso especial
    }
  }

  cout << "Puentes:\n";
  for(int i = 0; i < ABs.size(); i++)
    cout << ABs[i].first << " " << ABs[i].second << "\n";

  cout << "Puntos:\n";
  for(int i = 0; i < APs.size(); i++)
    if(APs[i]) cout << i << "\n";

  return 0;
}
```

References I