

Grafos

Grafos Bipartidos

Prof. Edson Alves

Faculdade UnB Gama

Grafos bipartidos

Grafos bipartidos

Um grafo é **bipartido** se todos os seus vértices podem ser coloridos usando apenas duas cores, de modo que todos os pares de vértices vizinhos tenham cores distintas.

Identificação de grafos bipartidos

Identificação de grafos bipartidos

- ★ Um grafo bipartido pode ser identificado por meio de uma travessia

Identificação de grafos bipartidos

- ★ Um grafo bipartido pode ser identificado por meio de uma travessia
- ★ Inicialmente não há atribuição de cores aos vértices

Identificação de grafos bipartidos

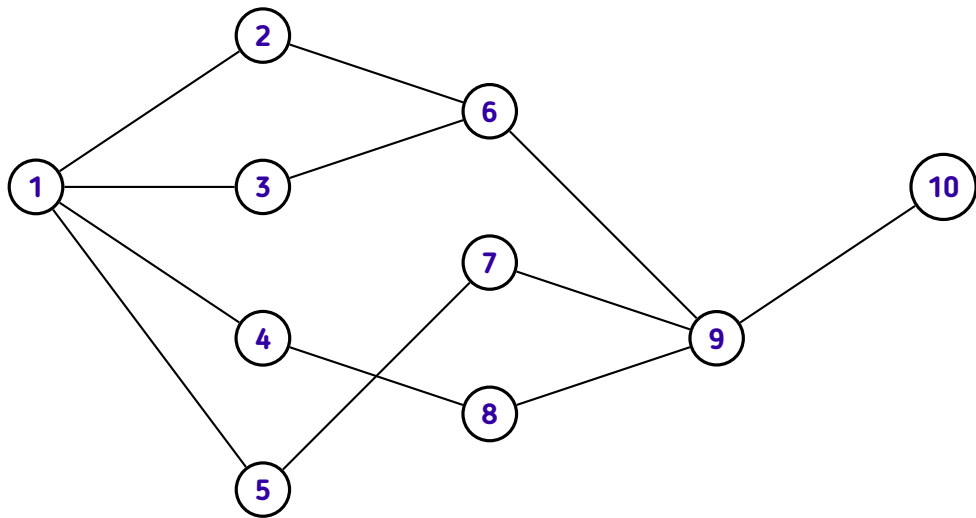
- ★ Um grafo bipartido pode ser identificado por meio de uma travessia
- ★ Inicialmente não há atribuição de cores aos vértices
- ★ O vértice de partida recebe então uma cor

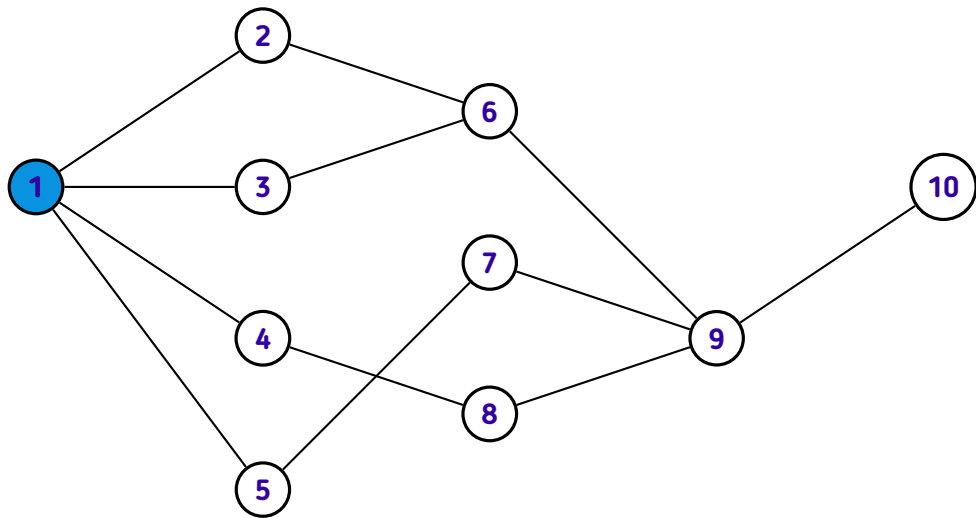
Identificação de grafos bipartidos

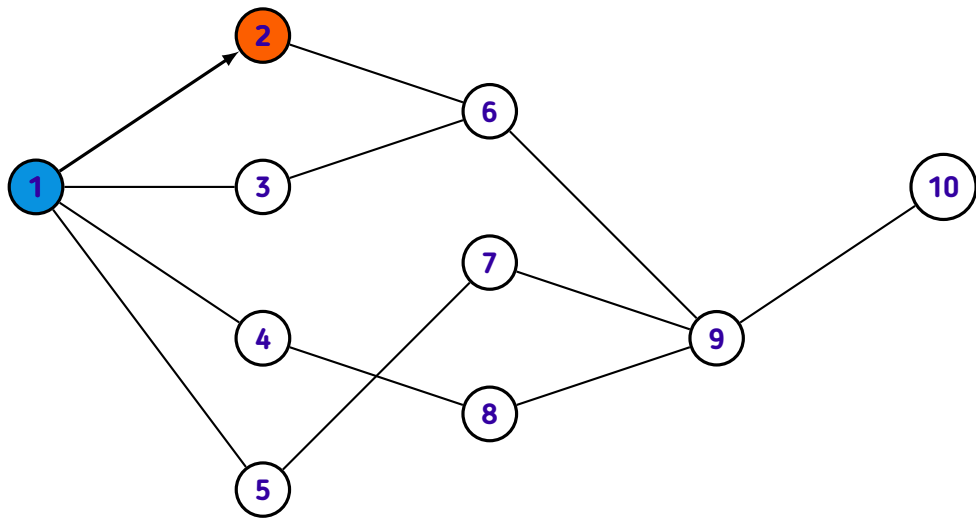
- ★ Um grafo bipartido pode ser identificado por meio de uma travessia
- ★ Inicialmente não há atribuição de cores aos vértices
- ★ O vértice de partida recebe então uma cor
- ★ A travessia continua nos seus vizinhos, que devem receber a cor oposta

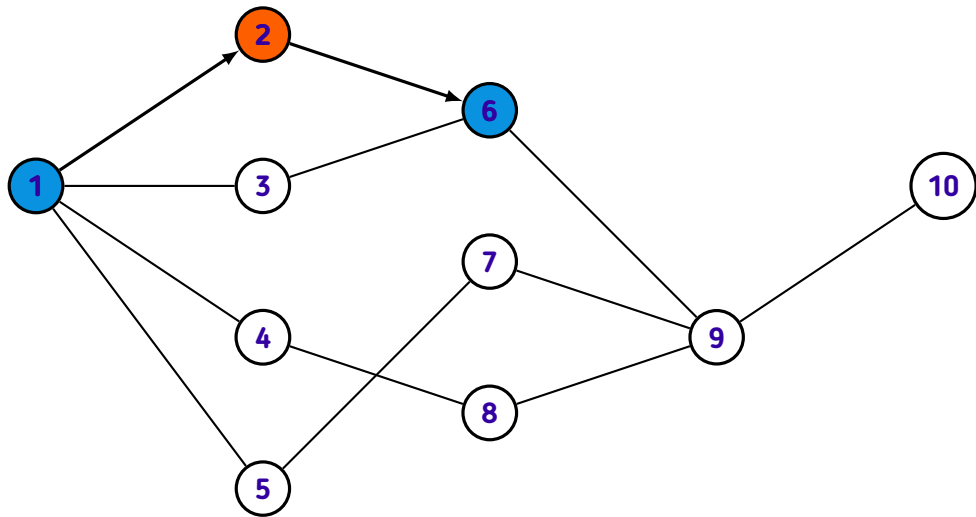
Identificação de grafos bipartidos

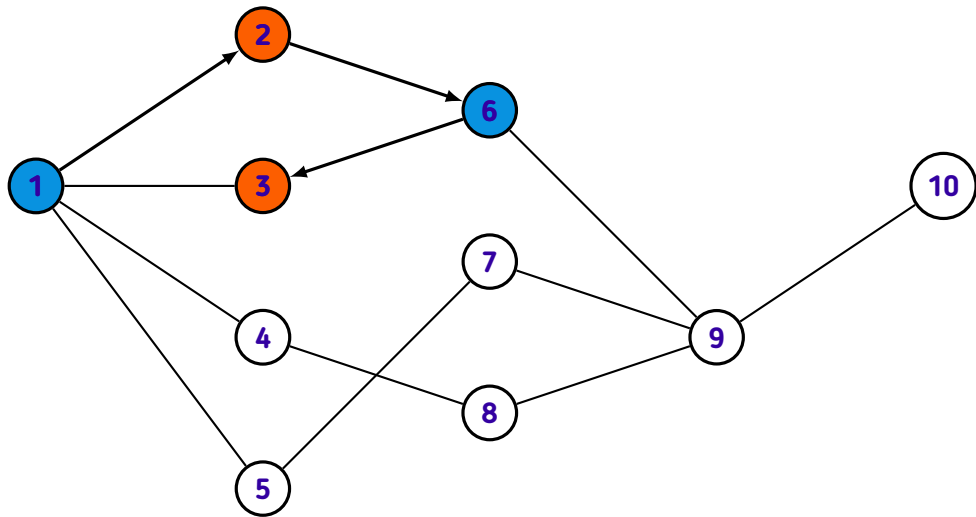
- ★ Um grafo bipartido pode ser identificado por meio de uma travessia
- ★ Inicialmente não há atribuição de cores aos vértices
- ★ O vértice de partida recebe então uma cor
- ★ A travessia continua nos seus vizinhos, que devem receber a cor oposta
- ★ Se a travessia atingir um nó já colorido, e a cor dele for a mesma do nó que o antecedeu na travessia, o grafo não é bipartido

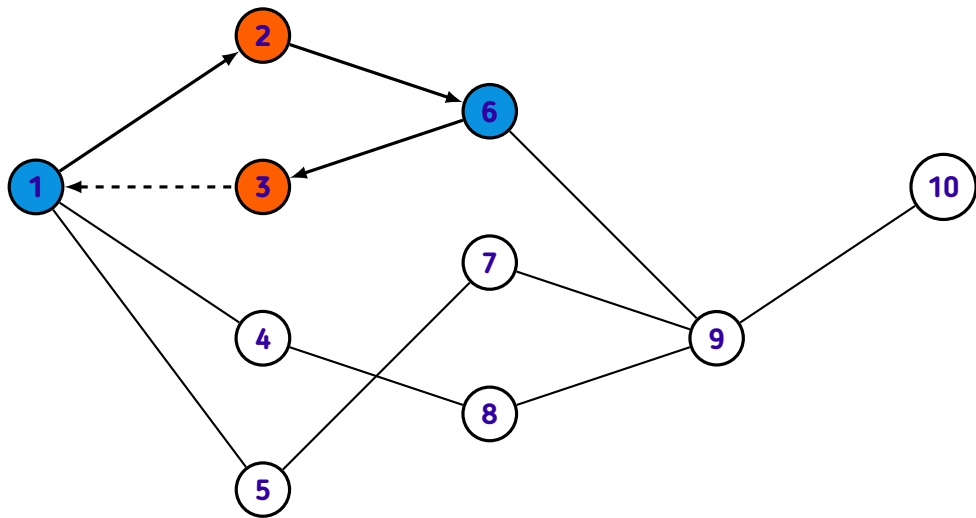


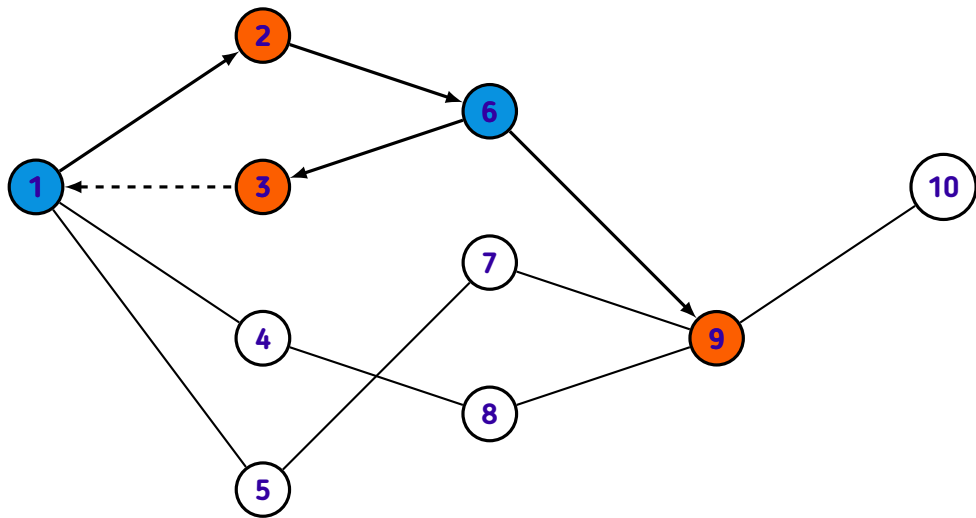


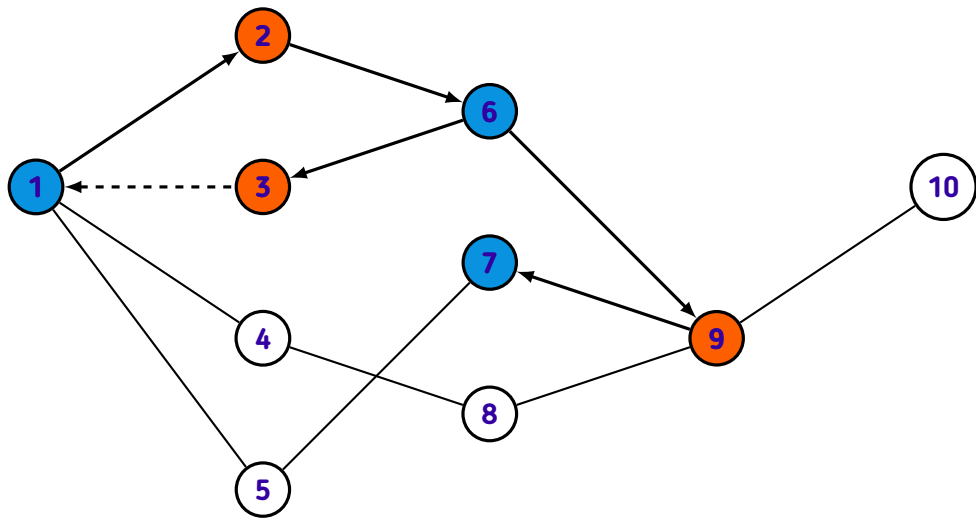


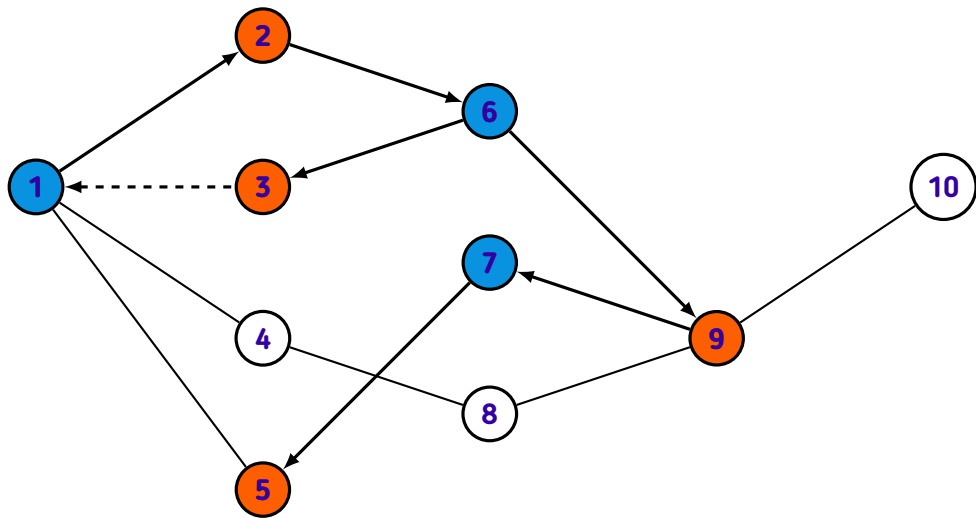


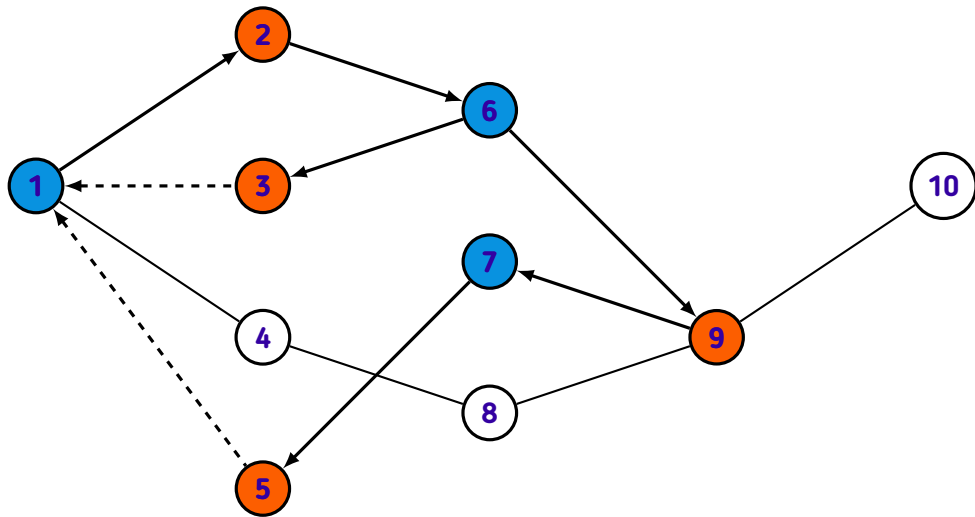


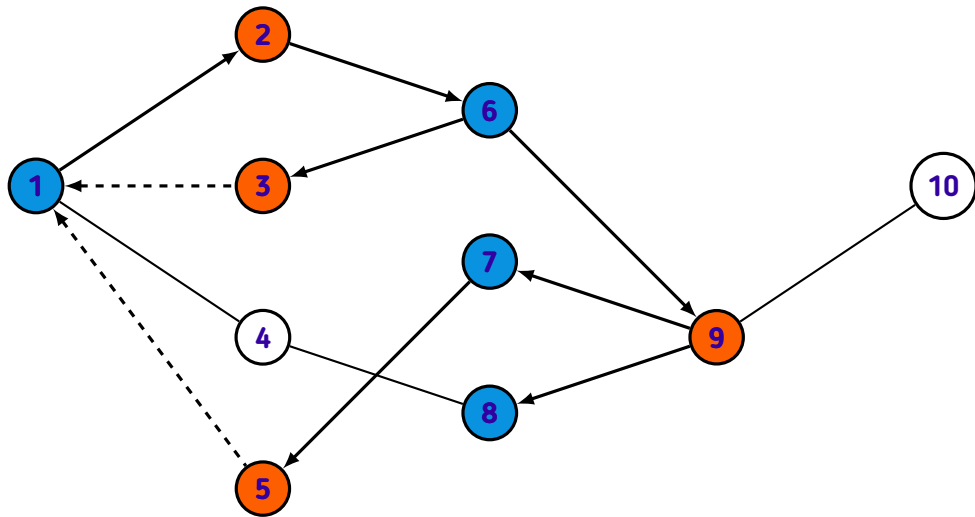


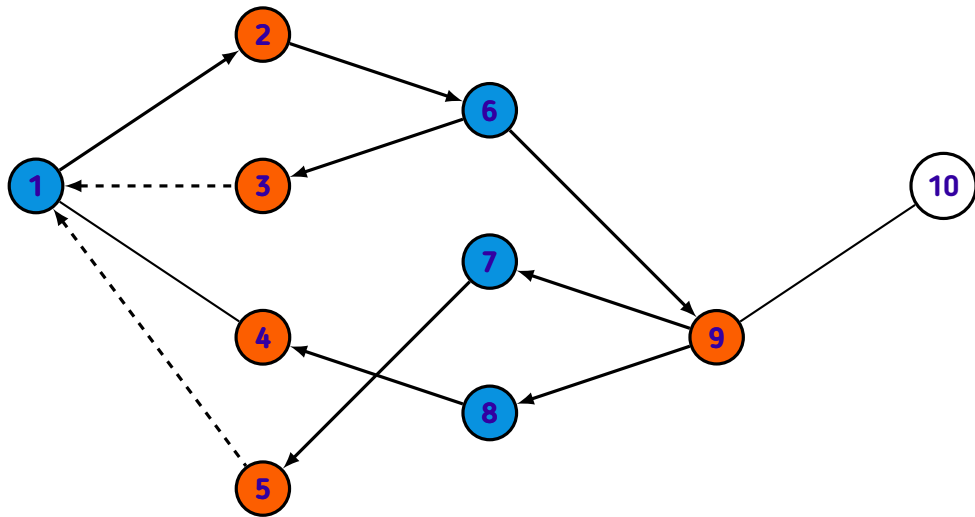


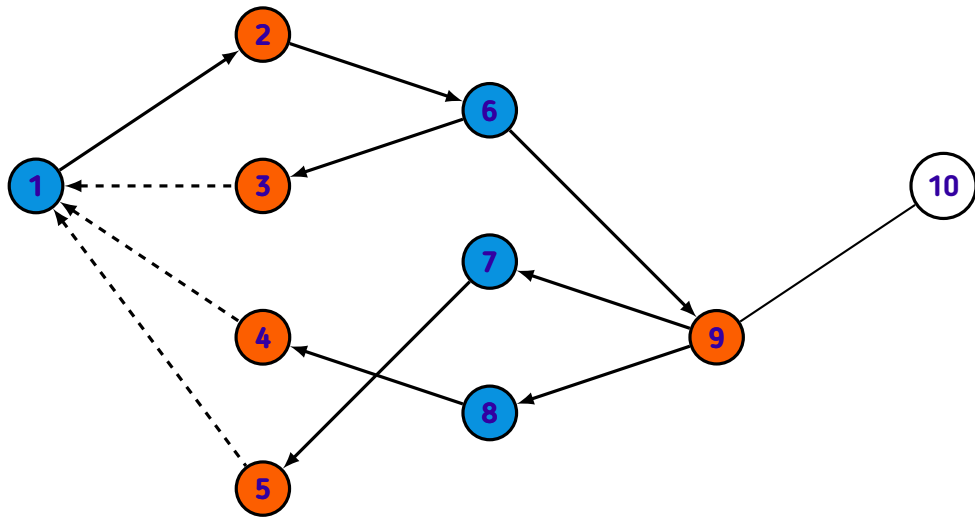


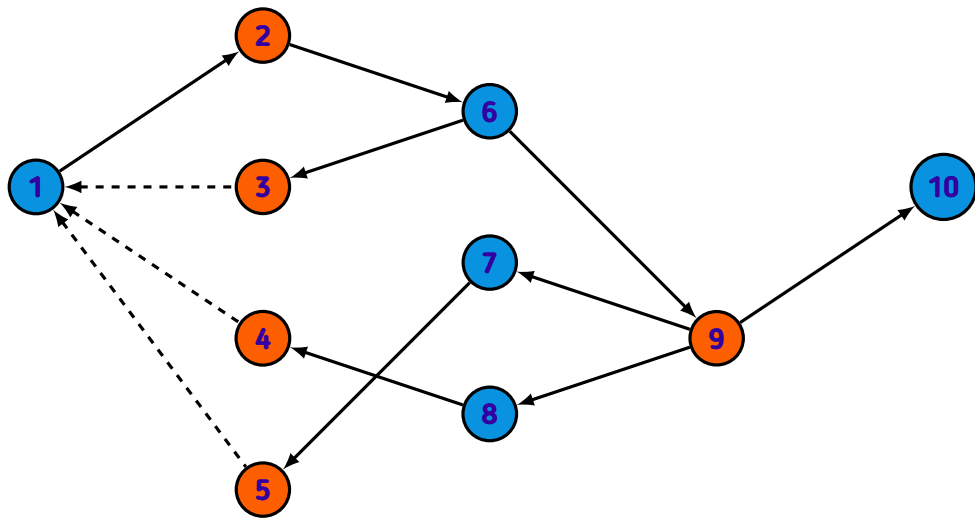












```
bool bfs(int s)
{
    const int NONE = 0, BLUE = 1, RED = 2;
    queue<int> q;
    q.push(s);
    color[s] = BLUE;

    while (not q.empty())
    {
        auto u = q.front(); q.pop();

        for (auto v : adj[u])
            if (color[v] == NONE) {
                color[v] = 3 - color[u];
                q.push(v);
            } else if (color[v] == color[u])
                return false;
    }

    return true;
}
```



```
bool is_bipartite(int N)
{
    for (int u = 1; u <= N; ++u)
        if (color[u] == NONE and not bfs(u))
            return false;

    return true;
}
```

Problemas sugeridos

1. [AtCoder Beginner Contest 131 – Problem F: Must Be Rectangular!](#)
2. [Codeforces Round #435 – Problem B: Mahmoud and Ehab and the bipartiteness](#)
3. [OJ 10505 – Montesco vs Capuleto](#)
4. [OJ 11080 – Place the Guards](#)

Referências

1. HALIM, Felix; HALIM, Steve. *Competitive Programming 3*, 2010.
2. LAAKSONEN, Antti. *Competitive Programmer's Handbook*, 2018.
3. SKIENA, Steven; REVILLA, Miguel. *Programming Challenges*, 2003.