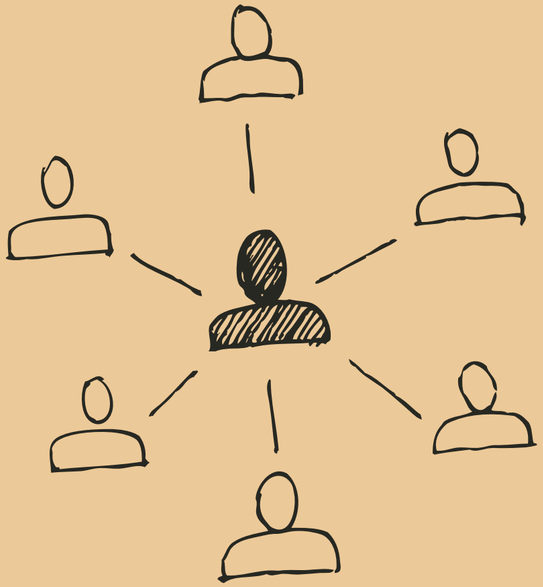
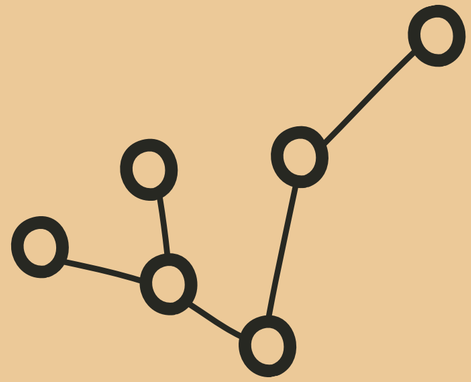
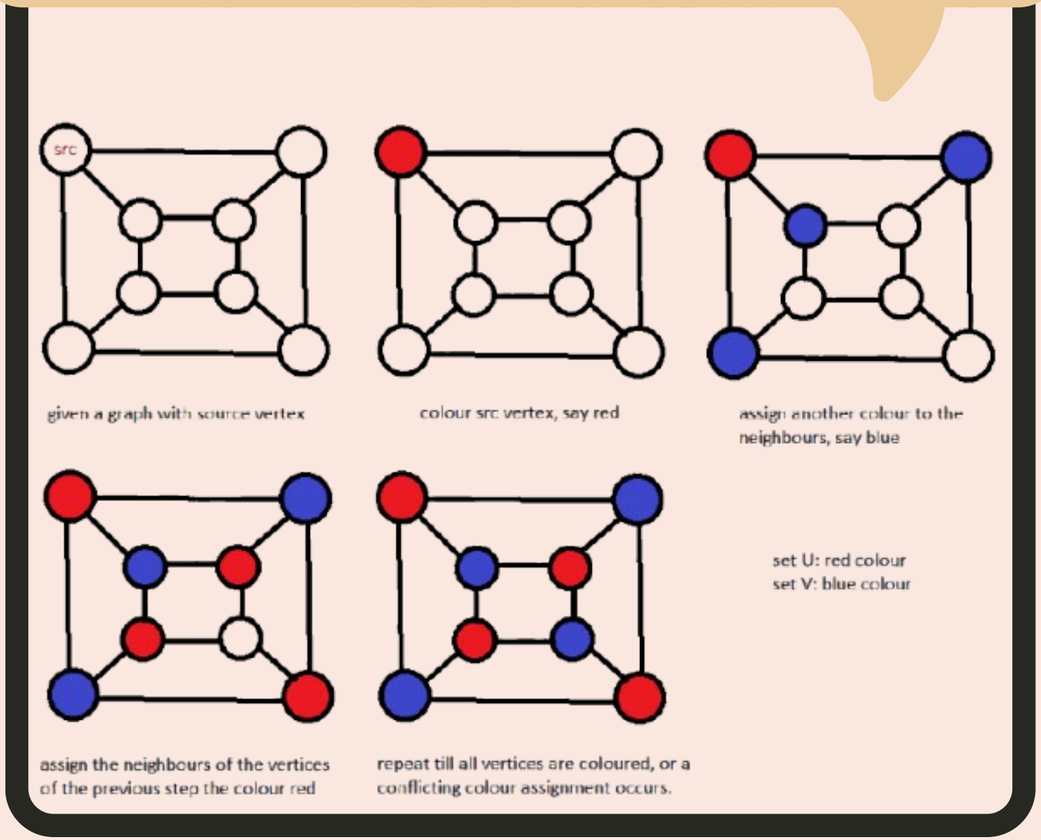
TEORIA DOS GRAFOS

Lukas Freitas de Carvalho - 202376033





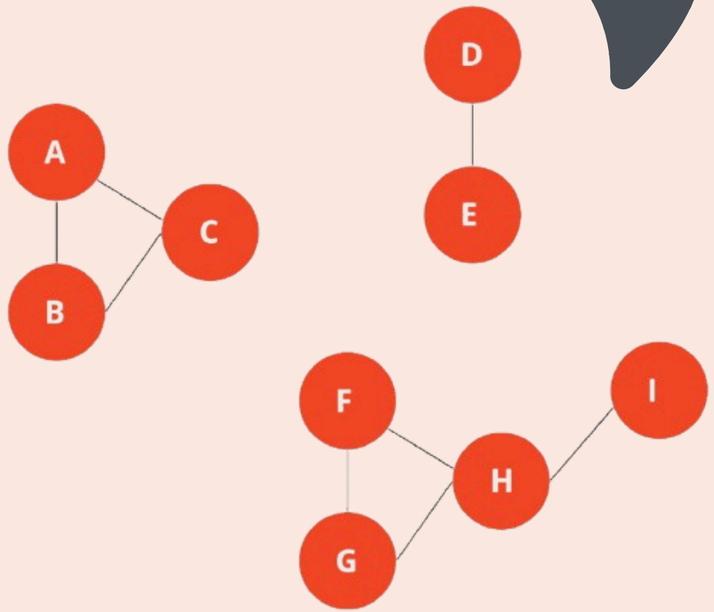
Coloração no grafo Bipartido




```cpp
bool Grafo_matriz::eh_bipartido() {
    int cor[getOrdem()];
    for (int i = 0; i < getOrdem(); i+=1) {
        cor[i] = -1;
    }

    int pilha[getOrdem()];
    int topo = -1;

    for (int i = 0; i < getOrdem(); i+=1) {
        if (cor[i] == -1) {
            pilha[++topo] = i;
            cor[i] = 0;

            while (topo >= 0) {
                int u = pilha[topo-=1];

                for (int v = 0; v < getOrdem(); v+=1) {
                    bool arestaExiste = false;

                    if (eh_direcionado()) {
                        arestaExiste = *retornaCelulaMatriz(u, v) != 0;
                    }
                    else if (u < v) {
                        arestaExiste = *retornaCelulaMatriz(u, v) != 0;
                    }

                    if (arestaExiste) {
                        if (cor[v] == -1) {
                            cor[v] = 1 - cor[u];
                            pilha[++topo] = v;
                        } else if (cor[v] == cor[u]) {
                            return false;
                        }
                    }
                }
            }
        }
    }

    return true;
}
```

```cpp
bool Grafo_lista::possuiArticulacao() {
    int nInicial = getNConexo();

    for (int i = 0; i < getOrdem(); i++) {
        NodeVertex* no = this->Vertice->getNodeById(i);
        if (!no || !no->getAtivo()) continue;

        bool ativoOrig = no->getAtivo();
        no->setAtivo(false);

        NodeEdge* aresta = no->getArestas()->getPrimeiro();
        while (aresta) {
            aresta->setAtivo(false);
            aresta = (NodeEdge*)aresta->getProx();
        }

        int nFinal = getNConexo();

        no->setAtivo(ativoOrig);

        aresta = no->getArestas()->getPrimeiro();
        while (aresta) {
            aresta->setAtivo(true);
            aresta = (NodeEdge*)aresta->getProx();
        }

        if (nFinal > nInicial) {
            return true;
        }
    }

    return false;
}
```

Algorítmo para identificar Articulações



OBRIGADO!