**Grafos**

Tabela é uma estrutura de dados e tem a mesma expressividade de frases. Relações entre objetos -> João gosta de Maria (FOL). Grafos descreve relação entre objetos. G = (V, E). Cada objeto tem um atributo (todas as combinações possíveis) -> V/F.

Nota: 0,7 MP + 0,3 MT

Grafo eureliano: consigo passar por uma única vez por cada aresta e voltar a origem. Árvore é um grafo sem signo. Centralidade de autovetor e de Katz em redes complexas. 7 pontes de Koningsberg

VER SLIDES:

* Grafo simples é um grafo que não possui arestas múltiplas(mais de uma para um msm ponto)
* Grafo trivial se for de ordem zero (vazio) ou um (um vértice)
* Um laço é uma aresta que começa e termina no mesmo vértice
* Adjacência: Vértices são extremos de uma mesma aresta
* Grau de um vértice é a quantidade de arestas incidentes de um vértice
* Grafo completo: todos seus vértices forem adjacentes. Possui n\*(n-1)/2 arestas
* Grafo simples é um grafo que não tem mais de uma arestas ligando dois vértices adjacentes
* Vértice fonte e sumidouro
* Quando o grafo é orientado, a gente divide em grau de entrada e saída
* Quando todos os vértices têm o mesmo grau o grafo é regular
* Arestas podem ter peso
* Caminho é uma sequência de vértices e arestas que une dois vértices
* Um caminho de K vértices, o comprimento é K - 1 do caminho (quando não tem peso)
* Caminho simples é quando não repete nenhum vértice
* Circuito é um caminho onde v(1) = v(k+1)
* Ciclo é um circuito onde todos os vértices são distintos, exceto o primeiro é último
* Árvore é um grafo acíclico e conexo. Árvore enraizada tem um nó raiz que os outros se afastam
* Floresta é um conjunto de árvores
* Hamiltoniano(passar por todos os vértices) e euleriano(um aresta uma vez)
* Grafo conexo sempre há um caminho pra qualquer vértice
* Todo euleriano é conexo e todos vértices possuem grau par (circuito)
* Grafo totalmente desconexo quando não possui arestas
* Grafo orientado é fortemente conexo quando de qualquer vértice eu consigo chegar a qualquer outro vértice
* Grafo bipartido(K 4,3) se considera a natureza de dois vértices
* Bipartido completo cada natureza de vértice está ligada a todos os vértices da outra natureza
* Complemento de um grafo são as arestas que deixam ele completo
* Isomorfismo é quando dois grafos são semelhantes entre si (não importa a distancia entre eles ou a posição dos vértices) -> preserva relação de incidência e o grau de cada vértice. Possuem a mesma matriz de adjacência
* Árvore geradora e subgeradora

M[i,j] = 0 ou 1 -> ligação de i (linha) a j (coluna) -> num grafo direcionado

Quando o grafo é não direcionado, a matriz de adjacência é simétrica

Quando o grafo tem peso, a gente usa o -1 para representar que não há ligação

Custo da matriz no grafo(representação/armazenamento): O(n²)

Custo achar aresta: O(1)

Cada matriz de adjacência me leva a um grafo, mas o contrário não (pode ter peso, pode ter -1, etc)

Faz a matriz transposta para saber se o grafo é fortemente conexo e ver se são iguais

Lista de adjacência: O(n+m)

Teste de aresta: O(grau do i)

Na prática (critério), um grado esparso tem a o número de arestas na mesma ordem de grandeza do que o vértice

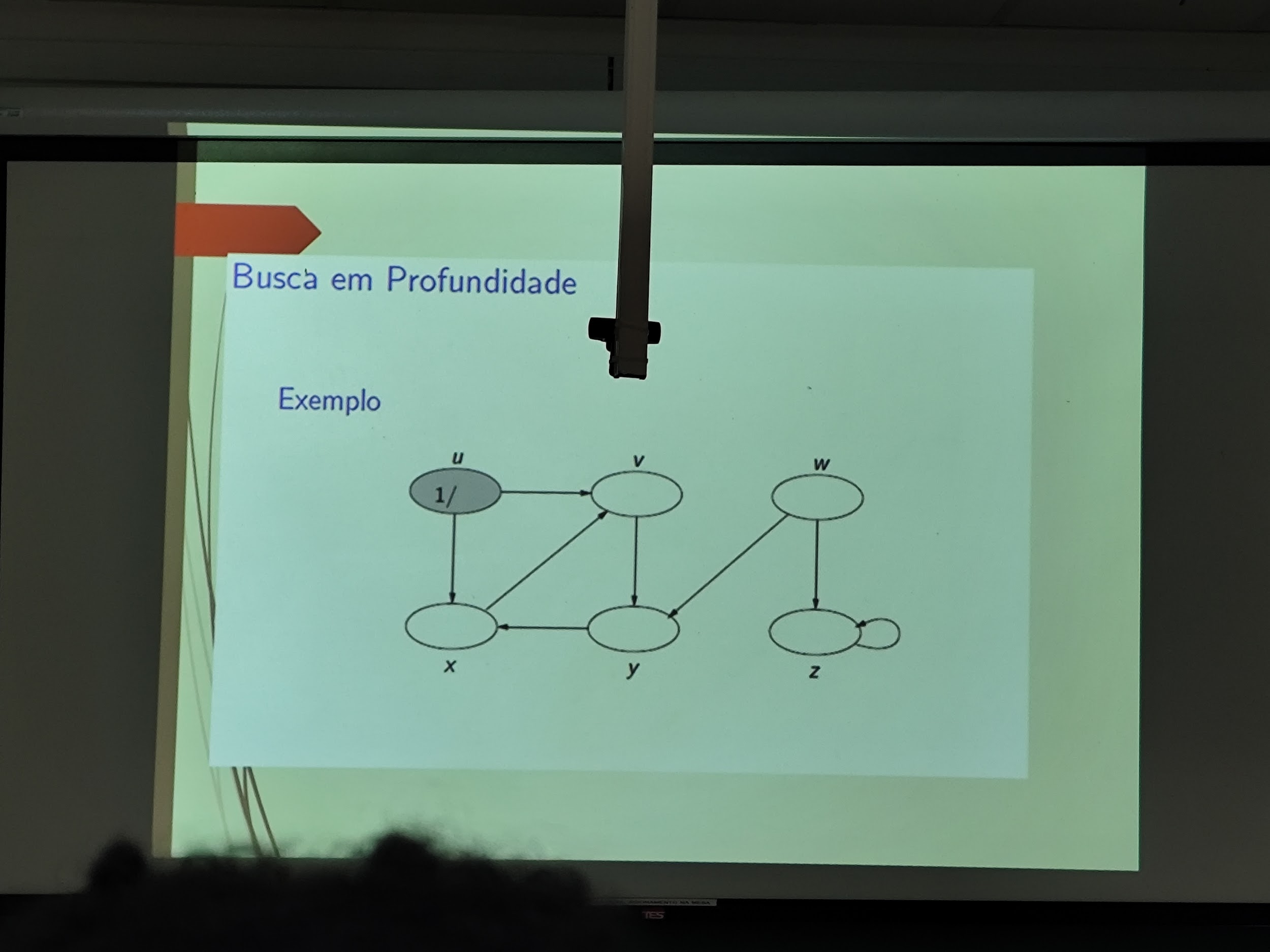
No nó de cabeçalho posso guardar características da lista, tipo grau do vértice

Grafos pequenos e densos, a matriz de adjacência é melhor. Só que a maioria dos casos os grafos são esparsos, por isso que é melhor lista de adjacência

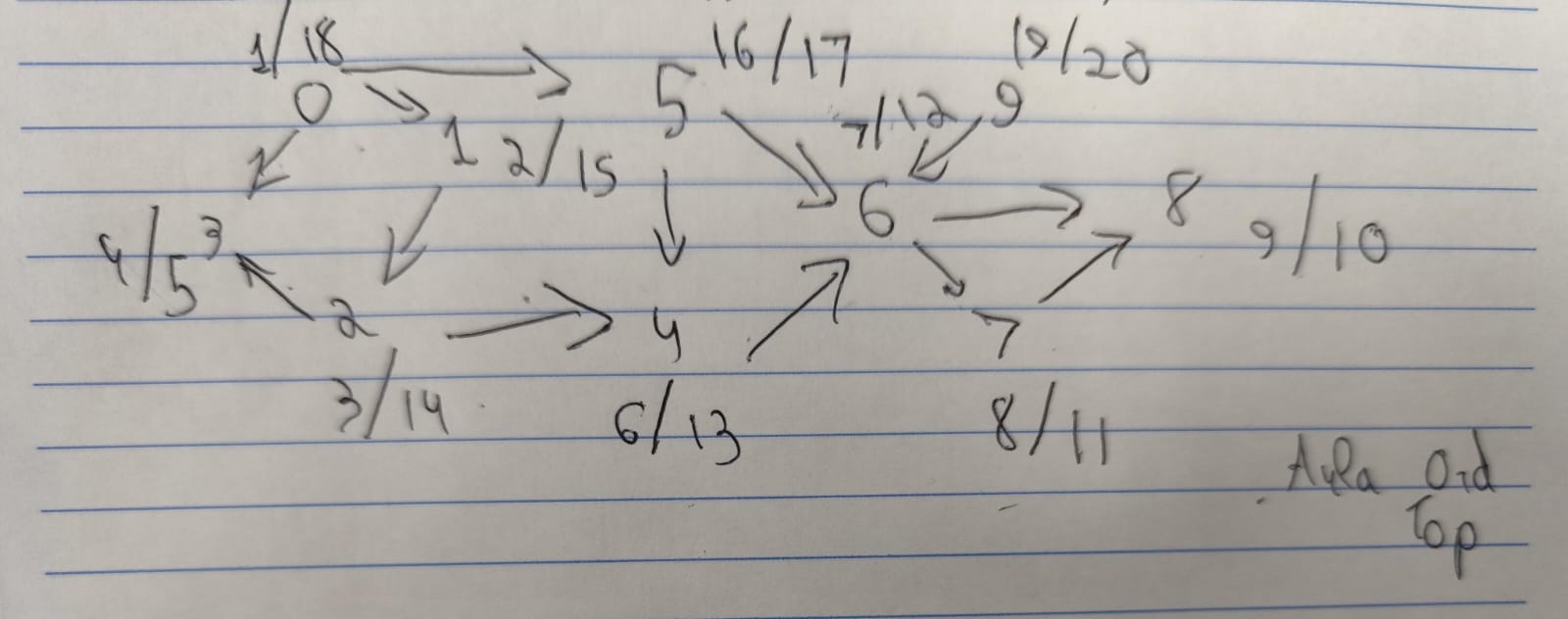
**BUSCA EM GRAFOS**

**-> Busca em Largura:**

Ver algoritmo. O vetor predecessor não mostra ele mesmo o caminho mínimo. Vc usa ela pra a partir do final ver quem precede quem de trás pra frente



Final fica assim:



u -> v -> y -> x

Algoritmo para encontrar componentes fortemente conexos de um grafo G (que é um grafo fortemente conexo dentro de um grafo). GAD: grafo assíncrono dirigido -> formado a partir do grafo de componentes fortemente conexo:

Duas bucas em profundidade: Uma no G e outra no G^T.

OBS: BFS gasta mais memória que DFS

