

```
void dfs(int node, boolean[] path, ArrayList<Integer> res) {
    path.add(node);
    psf.add(node);

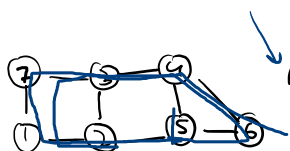
    int N = graph.size()-1; //number of nodes
    if(psf.size() == N){
        //this is hamiltonian path, add it to res
        System.out.println(psf);
        res.add(new ArrayList<Integer>(psf));

        psf.remove(psf.size()-1);
        path[node] = false;
        return;
    }

    for(int nbr: graph.get(node)){
        if(!path[nbr]){
            dfs(nbr, path, psf, graph);
        }
    }

    path[node] = false;
    psf.remove(psf.size()-1);
}
```

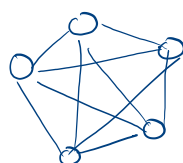
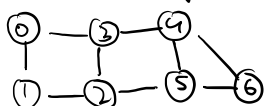
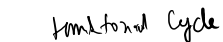
Hamiltonian Path



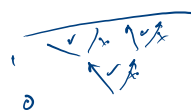
Any path that visits all nodes
exactly once.

$\frac{1}{1} \frac{1}{2} \frac{1}{3} \frac{1}{4} - \frac{1}{5} - \frac{1}{6} \frac{1}{7} 1234567$
 $\frac{1}{1} \frac{1}{2} \frac{1}{3} 4$

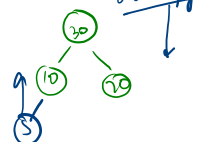
[5, 2, 1, 7, 3, 4, 6] Cycle



Callⁿ



Joan Hepp


$$\begin{aligned} & \mathcal{D}f(i, j, \text{grid}) \leq \\ & \mathcal{D}f(i, j, \text{end}) \text{ and } \mathcal{D}f(i, j, \text{start}) \\ & \text{grid}(i)(j) = 0, \\ & \mathcal{D}f(i-1, j-1) \\ & \mathcal{D}f(i, j-1, \text{end}), \mathcal{D}f(i, j-1, \text{start}) \\ & \mathcal{D}f(i-1, j, \text{end}), \mathcal{D}f(i-1, j, \text{start}) \end{aligned}$$
