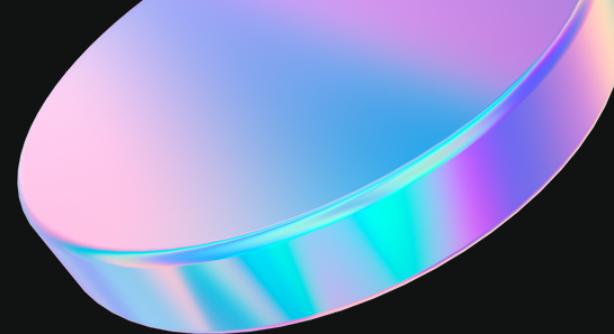
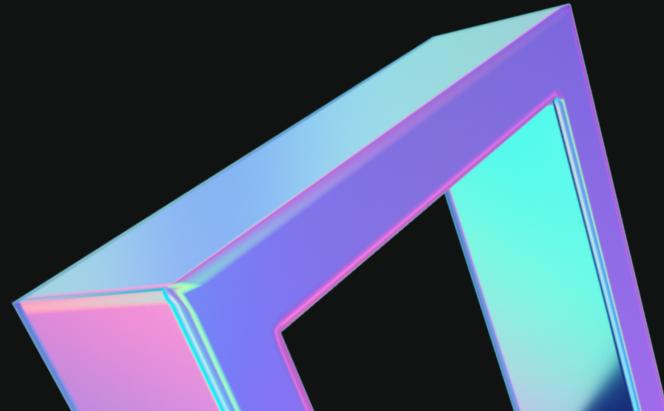
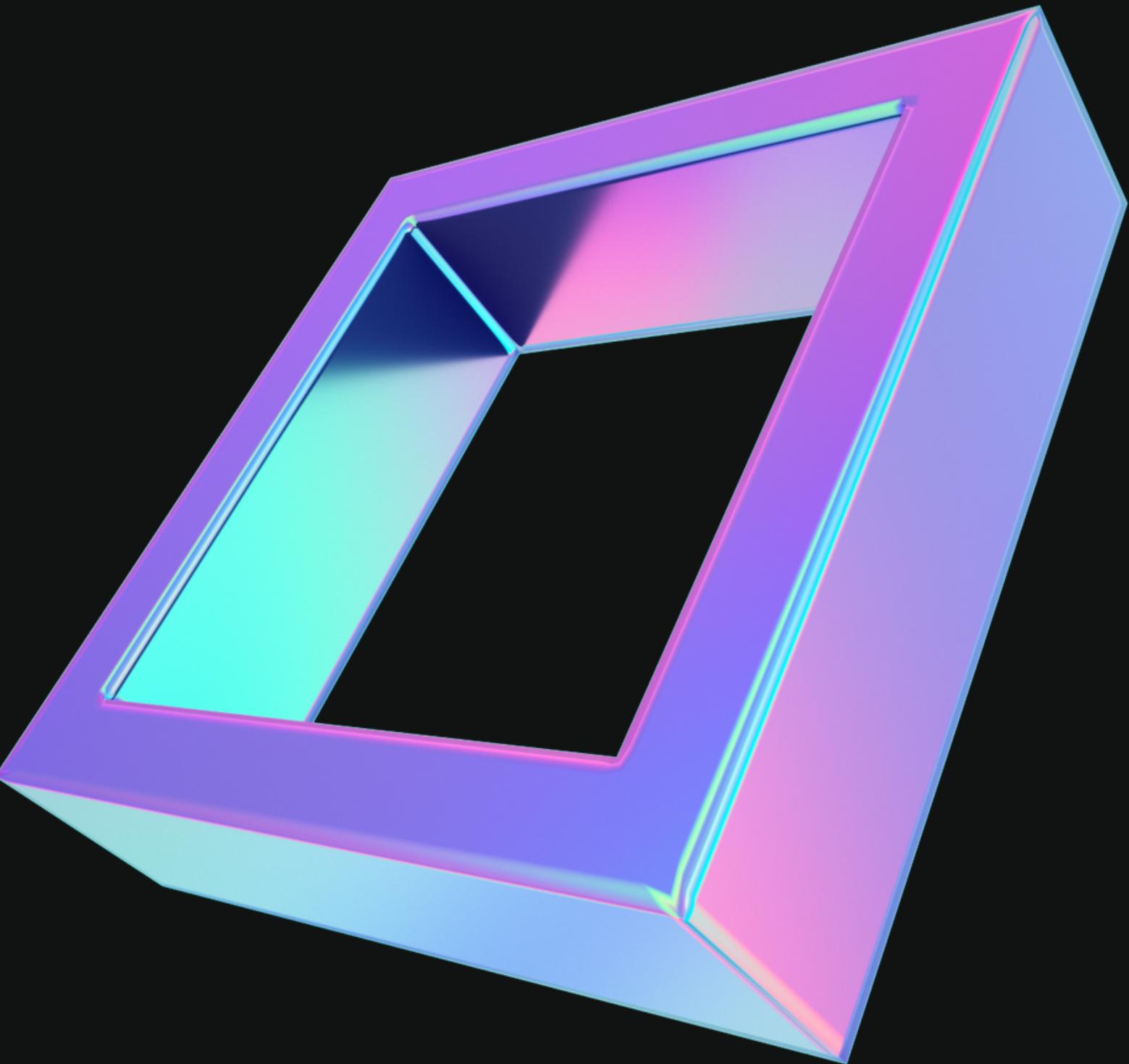


Graph Xpress

Pablo Montero Rollán | 112110



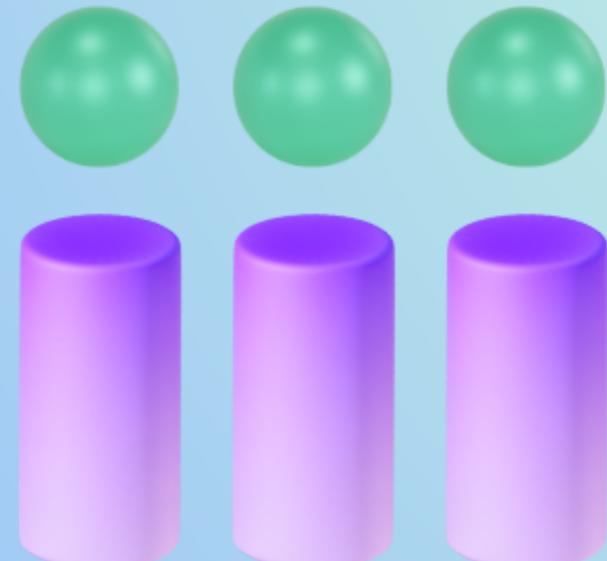
Inicial idea & Problems

1º Exercise

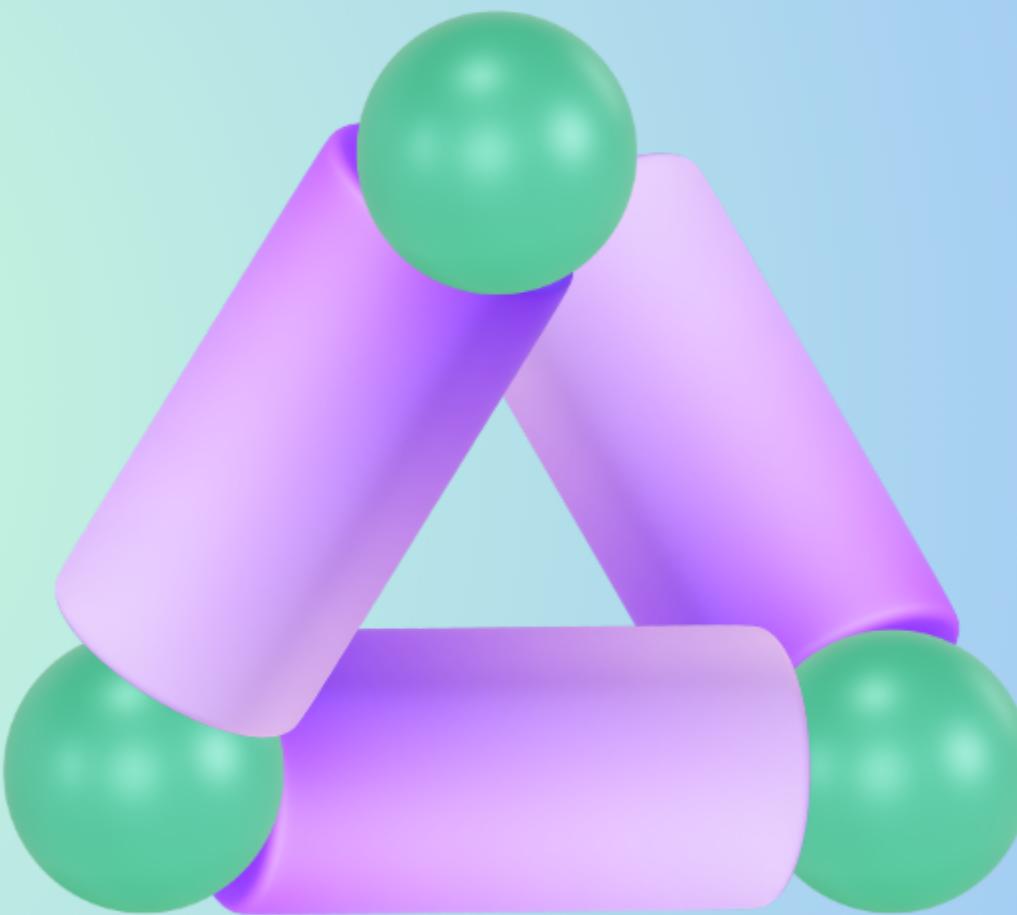
Enunciate

Given these vertices and edges, recreate a cycle graph 3

Vertices & edges



Result



Check

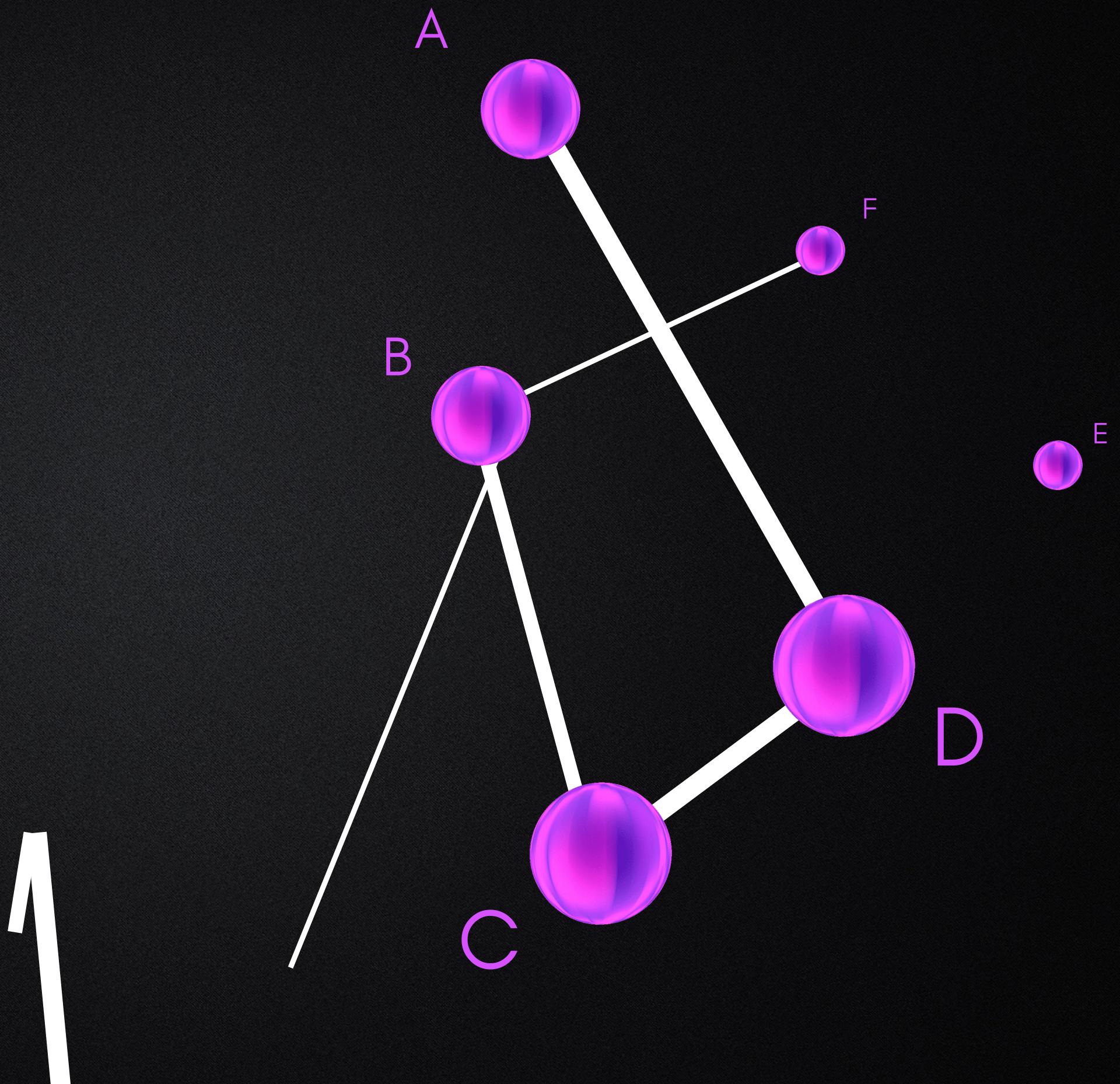
New idea

Title

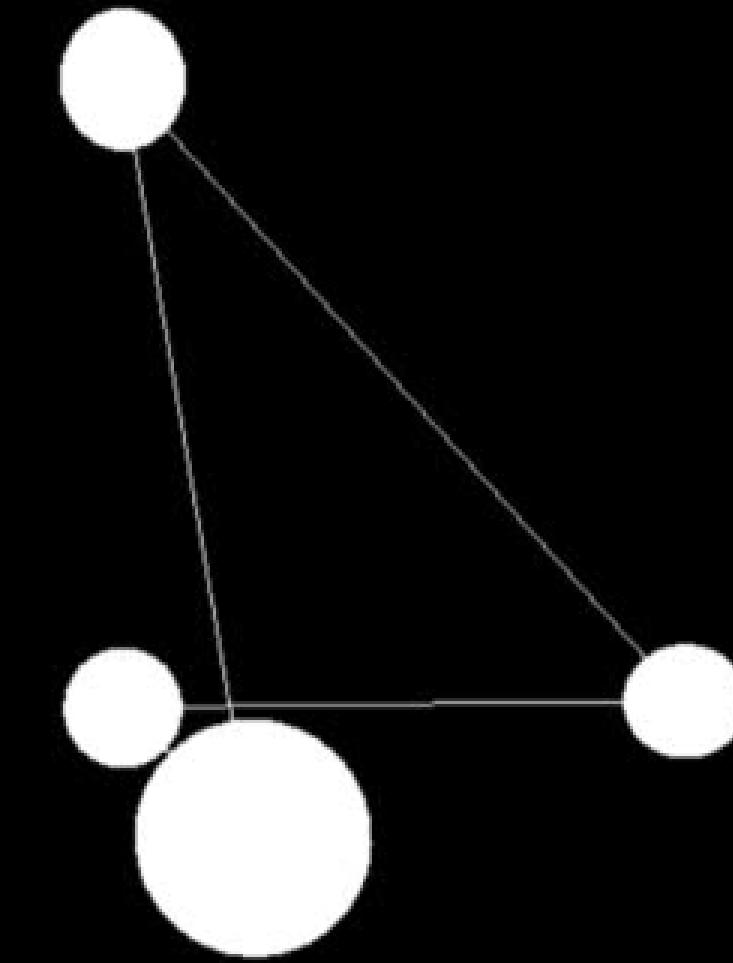
Vertices

Edges

Show



First version



Final version

GraphXpress

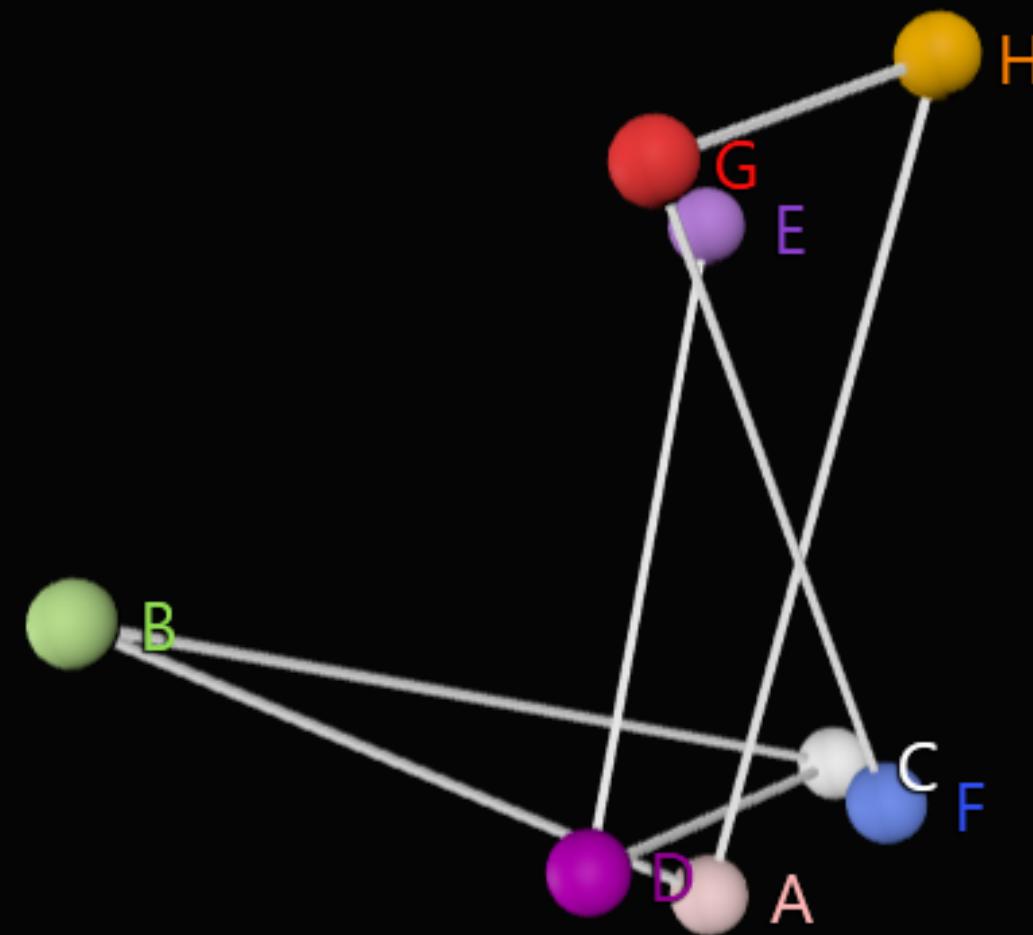
Vertixes

A,B,C,D,E,F,G,H

Edges

AB,BC,CD,DE,FG,G

Revisar



Code

Cromatic number

```
function numeroCromatico(aristas) {  
    const lista = aristas.split(','');  
  
    const frecuenciaLetras = {};  
  
    for (const elemento of lista) { ... }  
  
    for (const letra in frecuenciaLetras) { ... }  
  
    return false;  
}  
  
export {numeroCromatico};
```

Graph generator

```
import generarCoordenadas from '/js/coordenadas.mjs';

function generarArchivoMol(verticesInput, aristasInput, elementos) {
    let vertices = verticesInput.split(',');
    let adyacencias = aristasInput.split(',');
    let texto = '';
    let numeroDeAtomos = vertices.length;
    let numeroDeEnlaces = adyacencias.length;
    let radio = 2;

    // HEADER
    texto += `HEADER      NONAME 16/5/2023
NONE\n`;

    texto += 'TITLE
texto += 'AUTHOR      Chemical Structure Services at http://cactus.nci.nih.gov
texto += 'REVDAT      1 23-Apr-10  0
NONE\n';
NONE\n';
NONE\n';

    // ATOM
    let coordenadas = generarCoordenadas(numeroDeAtomos, radio);
    for (let i = 0; i < numeroDeAtomos; i++) {
        }

    // CONECT
        for (let i = 0; i < numeroDeEnlaces; i++) {
            }

    // END
    texto += 'END ';

    console.log(texto);
    return texto;
}

export { generarArchivoMol };
```

Coordinates generator

```
function generarCoordenadas(numAtomos, radio) {  
    const coordenadas = [];  
  
    const incrementoAngular = Math.PI * (3 - Math.sqrt(5));  
    const offset = 2 / numAtomos;  
  
    for (let i = 0; i < numAtomos; i++) { ...  
    }  
  
    return coordenadas;  
}  
  
export default generarCoordenadas;
```

Work with addons

```
<script type="module">

    import * as THREE from 'three';
    import { generarArchivoMol } from './js/pdbgenerator.mjs';

    import { TrackballControls } from 'three/addons/controls/TrackballControls.js';
    import { PDBLoader } from 'three/addons/loaders/PDBLoader.js';
    import { CSS2DRenderer, CSS2DObject } from 'three/addons/renderers/CSS2DRenderer.js';

    let camera, scene, renderer, labelRenderer;
    let controls;

    let root;

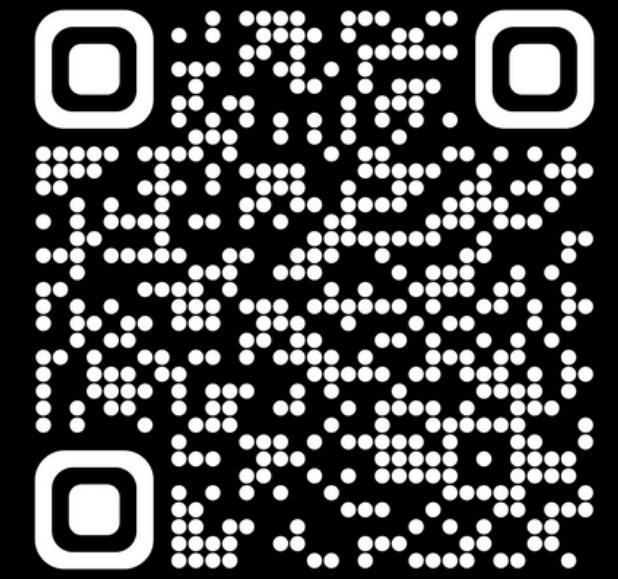
    const verticesInput = localStorage.getItem('verticesValue');
    const aristasInput = localStorage.getItem('aristasValue');
    const elementos = ['B', 'F', 'H', 'I', 'K', 'N', 'O', 'P', 'S', 'U', 'Y'];
    const texto = generarArchivoMol(verticesInput, aristasInput, elementos);
    const url = URL.createObjectURL(new Blob([texto], { type: 'chemical/x-pdb' }));

    const loader = new PDBLoader();
    const offset = new THREE.Vector3();
```

Camera & Lights

```
function init() {  
  
    scene = new THREE.Scene();  
    scene.background = new THREE.Color( 0x050505 );  
  
    camera = new THREE.PerspectiveCamera( 70, window.innerWidth / window.innerHeight, 1, 5000 );  
    camera.position.z = 1000;  
    scene.add( camera );  
  
    const light1 = new THREE.DirectionalLight( 0xffffff, 0.8 );  
    light1.position.set( 1, 1, 1 );  
    scene.add( light1 );  
  
    const light2 = new THREE.DirectionalLight( 0xffffff, 0.5 );  
    light2.position.set( - 1, - 1, 1 );  
    scene.add( light2 );  
  
    root = new THREE.Group();  
    scene.add( root );
```

Demo



Link