

Competència Transversal EDA

- Aquest exercici s'ha de lliurar a través del Racó.
- Poseu el vostre nom complet i número de DNI.
- Contesteu a totes les qüestions en el propi full de l'enunciat.
- Quan doneu una referència a una font (llibre, revista, web, etc.), seguiu la norma "ISO-690 (author-date, English)". Podeu generar les referències en aquest format a:
www.citethisforme.com/guides/iso690-author-date-en
- Quan doneu una URL, si us plau escriviu **clarament** i useu, per exemple:
<https://shorturl.at>

(a) Digueu quin problema resol l'algorisme de Floyd-Warshall.

El problema que intenta resoldre aquest algorisme és el de trobar el camí més curt entre tots els parells de vèrtexs en un graf. És un algorisme dinàmic on es poden tractar grafs amb pesos o sense ('directed' o 'undirected'), però no funciona amb grafs amb cicles negatius (la suma dels pesos en un cicle és negativa). L'algorisme el que fa per dintre és agafar dos nodes, un sortida i l'altre destí (primer i segon bucle 'for'), i després va comprovant amb tots els nodes restants si hi ha algun camí més curt del que ja hi havia (tercer 'for').

(b) Quin és el seu cost asimptòtic en temps i espai en el cas pitjor?

En cas pitjor, el cost asimptòtic en temps és de $O(V^3)$, ja que el graf té V vèrtexs, i l'algorisme té 3 bucles 'for' niats on es recorren tots els vèrtexs. En el cas de l'espai, el cost asimptòtic és de $O(V^2)$, perquè es crea una matriu amb V files i V columnes per guardar la distància mínima entre cada parell de nodes.

(c) Sigui INF una constant entera de valor suficientment gran. La matriu de pesos M d'un graf dirigit amb pesos $G = (\{0, \dots, n-1\}, E)$ és la matriu $n \times n$ definida així: M_{uv} és el pes de l'aresta (u, v) si $(u, v) \in E$, i INF altrament.

Implementeu en C++ una funció

```
vector<vector<int>> floyd_warshall(const vector<vector<int>>& M);
```

que, donada la matriu de pesos d'un graf dirigit amb pesos, retorni la matriu obtinguda en aplicar-hi l'algorisme de Floyd-Warshall.

```
vector<vector<int>> floyd_warshall(const vector<vector<int>>& M) {  
    int n = M.size(); //obtenim nombre de vèrtexs  
    vector<vector<int>> distancia = M;  
  
    for (int i = 0; i < n; ++i) {  
        for (int j = 0; j < n; ++j) {  
            for (int k = 0; k < n; ++k) {  
                if (distancia[i][j] > distancia[i][k] + distancia[k][j] and  
                    distancia[i][k] != INF and distancia[k][j] != INF) {  
                    distancia[i][j] = distancia[i][k] + distancia[k][j];  
                }  
            }  
        }  
    }  
    return distancia;  
}
```

(d) Citeu les fonts que heu fet servir per contestar a les qüestions anteriors.

- Grafos - software para la construcción, edición y análisis de grafos. algoritmo_floyd_warshall [Grafos - software para la construcción, edición y análisis de grafos.] (Consulta: 18 de desembre 2024). Disponible a: <https://shorturl.at/fR4qS>
- Comment et al. (2024) Floyd Warshall algorithm, GeeksforGeeks. (Consulta: 18 de desembre 2024). Disponible a: <https://shorturl.at/qzcpM>
- Wikimedia, C. de los proyectos (2024) Algoritmo de Floyd-Warshall, Wikipedia, la enciclopedia libre (Consulta: 18 de desembre 2024) Disponible a: <https://shorturl.at/SzSjl>
- Ruedas, I.M.& L. (2021) Algoritmo de Floyd-Warshall, Medium (Consulta: 18 de desembre 2024). Disponible a: <https://shorturl.at/zehgl>