

GRAFOS VALORADOS

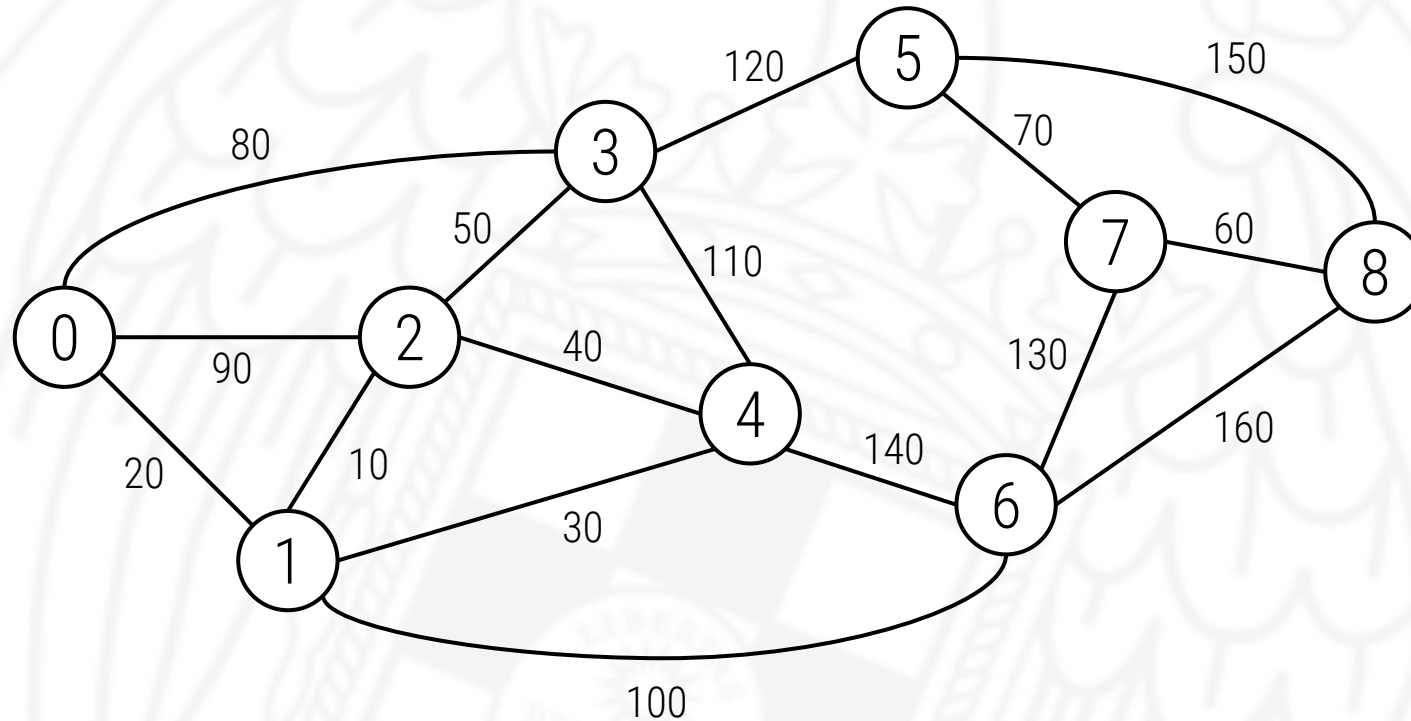


U N I V E R S I D A D
COMPLUTENSE
M A D R I D

ALBERTO VERDEJO

Grafos valorados

- ▶ Grafos cuyas aristas tienen asociado un valor (peso, coste).



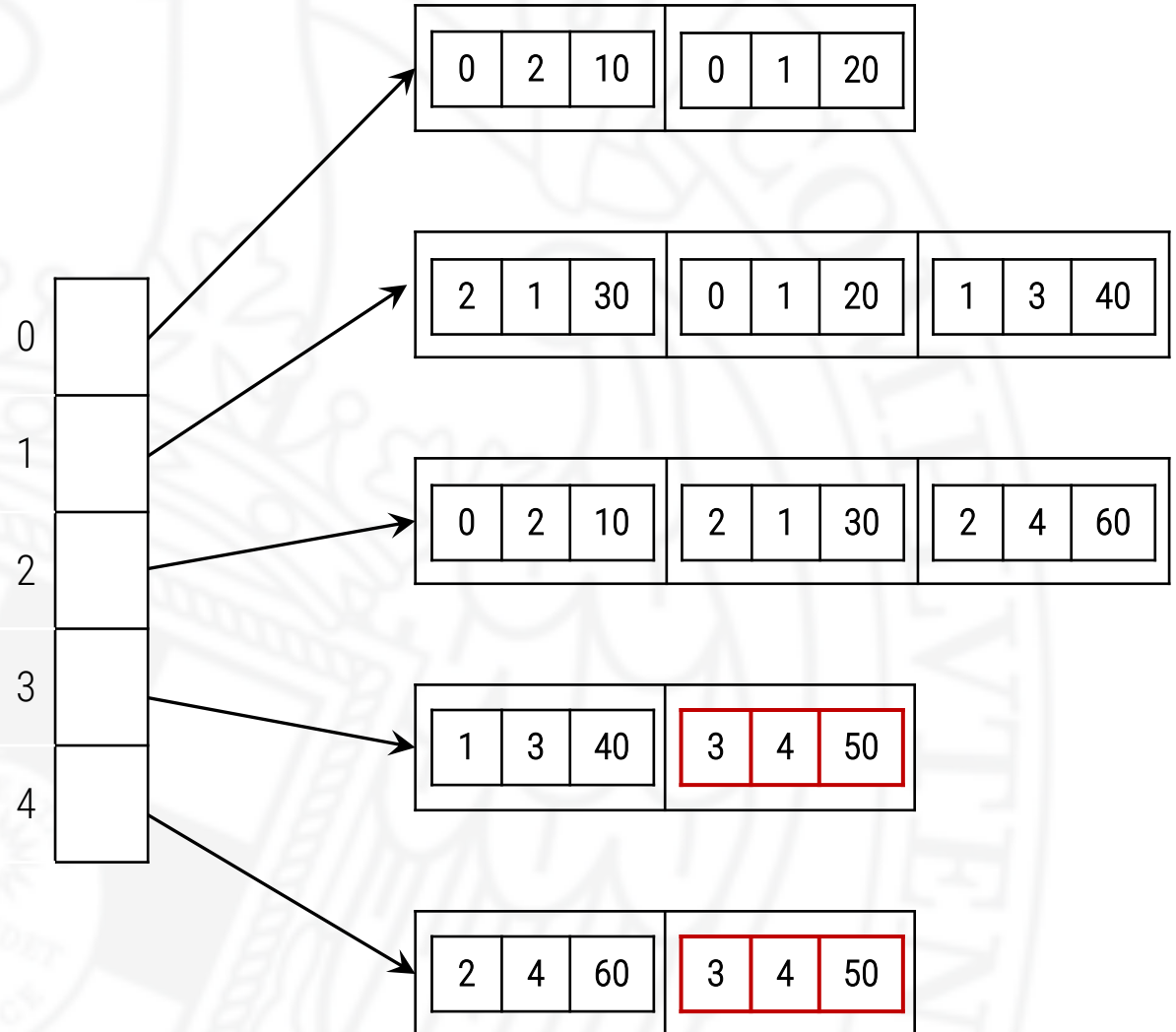
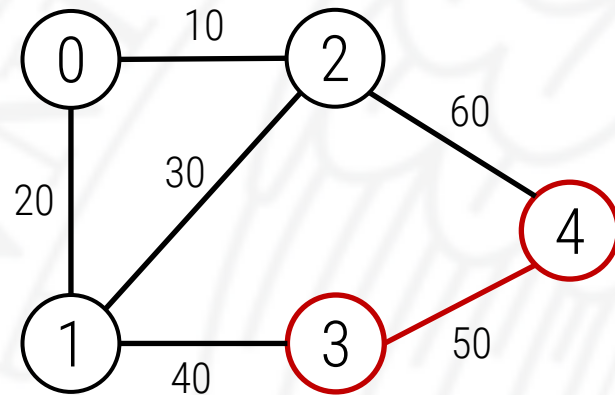
```
template <typename Valor>
class Arista {
public:
    Arista(int v, int w, Valor valor);
    int uno() const;
    int otro(int u) const;
    Valor valor() const;
    bool operator<(Arista<Valor> const& b) const;
    bool operator>(Arista<Valor> const& b) const;
};

    int v = arista.uno(), w = arista.otro(v);
```

Listas de adyacencia



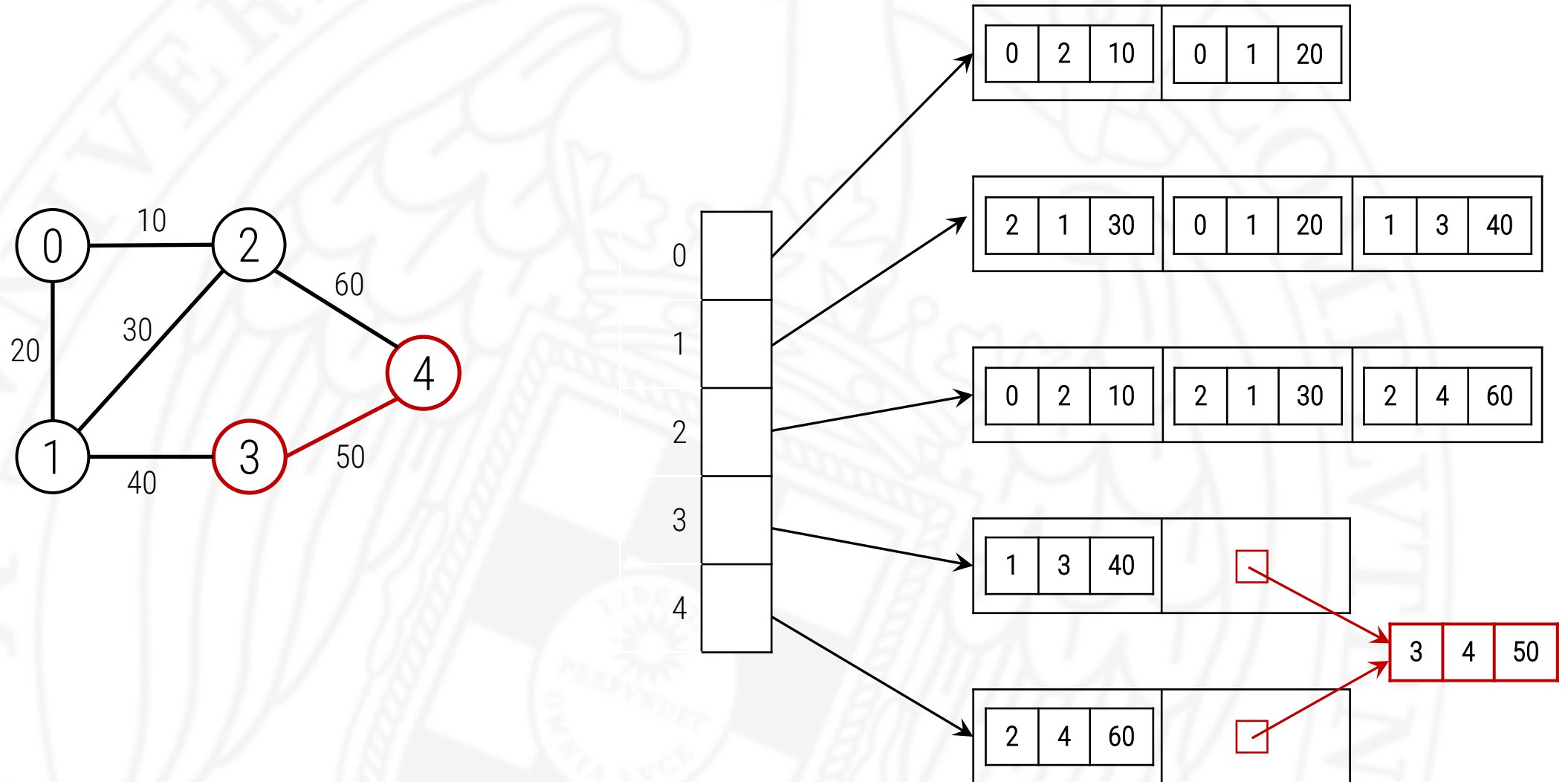
GrafoValorado.h



Listas de adyacencia



GrafoValorado.h



Grafo valorado



GrafoValorado.h

```
template <typename Valor>
class GrafoValorado {
public:
    GrafoValorado(int V);
    void ponArista(Arista<Valor> arista);
    int V() const;
    int A() const;
    AdysVal<Valor> const& ady(int v) const;
    std::vector<Arista<Valor>> aristas() const;
};
```



```
void ponArista(Arista<Valor> arista) {  
    int v = arista.uno(), w = arista.otro(v);  
    if (v < 0 || v >= _V || w < 0 || w >= _V)  
        throw std::invalid_argument("Vertice inexistente");  
    ++_A;  
    _ady[v].push_back(arista);  
    _ady[w].push_back(arista);  
}
```

```
std::vector<Arista<Valor>> aristas() const {  
    std::vector<Arista<Valor>> ars;  
    for (int v = 0; v < V(); ++v)  
        for (auto arista : ady(v))  
            if (v < arista.otro(v))  
                ars.push_back(arista);  
    return ars;  
}
```


Construcción de un grafo valorado

```
bool resuelveCaso() {  
    int V, A;  
    cin >> V >> A;  
    GrafoValorado<int> grafo(V);  
    int u, v, valor;  
    for (int i = 0; i < A; ++i) {  
        cin >> u >> v >> valor;  
        grafo.ponArista({u, v, valor});  
    }  
    ...  
}
```

nº de vértices
nº de aristas
5
12
0 2 10
3 1 30
2 4 20
4 3 15
...
extremos
valor

Recorrido en profundidad

```
// visita los nodos alcanzables desde v respetando el umbral
void dfs(GrafoValorado<int> const& G, int v, int umbral) {
    visit[v] = true;
    for (auto a : G.ady(v)) {
        if (a.valor() < umbral) {
            int w = a.otro(v);
            if (!visit[w])
                dfs(G, w, ancho);
        }
    }
}
```