

Problema 25

SOTILLO SOTILLO, CLARA (TAIS96)

ID envío	Usuario/a	Hora envío	Veredicto
59478	TAIS96	2021-10-19 10:16	AC
59412	TAIS96	2021-10-19 09:51	AC
59406	TAIS96	2021-10-19 09:49	WA
59395	TAIS96	2021-10-19 09:42	WA

Fichero Source.cpp

- *
 - * Nombre y Apellidos:
 - * Carlos Gomez Lopez TAIS45
 - * Clara Sotillo Sotillo TAIS96
- *
 - * El coste en tiempo en el caso peor de esta funcion es $O(A \log A)$ siendo A el numero de aristas con coste menor al del construir un aeropuerto del grafo, ya que el coste de calcular arm es $O(A \log A)$
 - * El coste en espacio adicional es de $O(A)$

Hemos creado un grafo valorado en el cual no hemos añadido las aristas con valor mayor o igual al coste de construir un aeropuerto.

Hemos creado una clase aeropuerto en la que realizamos el algoritmo de kruskal para hayar el coste minimo de construccion de carreteras.

Del conjunto disjunto creado para implementar kruskal sacamos el numero de conjuntos, que equivale al numero de componentes conexas del grafo y por tanto al numero de aeropuertos a construir.

Para finalizar el coste min lo sumamos al numero de aeropuertos necesarios el cual multiplicamos por su coste.

```
class aeropuerto {
public:
    aeropuerto(GrafoValorado<int> const& g) : coste(0), cd(g.V()), cont(0) {
        kruskal(g);
        cont = cd.num_cjtos();
    }
    int valcoste()
    {
        return coste;
    }
    int valcont() {
        return cont;
    }
private:
    int coste, cont ;
}
```

```

ConjuntosDisjuntos cd;
vector<Arista<int>> arm;
void kruskal(GrafoValorado<int> const& g) {
    auto arista = g.aristas();
    priority_queue<Arista<int>, vector<Arista<int>>, greater<Arista<int>>> queue(arista.
begin(), arista.end());

    while (!queue.empty()) {
        auto a = queue.top();
        queue.pop();

        int v = a.uno(), w = a.otro(v);
        if (!cd.unidos(v, w)) {
            cd.unir(v, w);
            arm.push_back(a);
            coste += a.valor();
            if (arm.size() == g.V() - 1) break;
        }
    }
}

};

bool resuelveCaso() {
    int n, m, aero;

    cin >> n >> m >> aero;

    if (!std::cin) // fin de la entrada
        return false;

    GrafoValorado <int> g(n);

    int a, b, val;

    for (int i = 0; i < m; i++) {
        cin >> a >> b >> val;
        if(val<aero) g.ponArista({ a-1,b-1,val });
    }

    aeropuerto A(g);

    cout << A.valcoste() + aero*A.valcont()<< " " << A.valcont() << "\n";

    return true;
}

```

