Practica 3 - Greedy

Raúl Granados López
Hossam El Amraoui Leghzali
Javier Montaño Rubio

Diseño de las componentes Greedy

- **Lista de candidatos:** Las aristas del grafo.
- **Lista de candidatos utilizados:** Las aristas que se han ido seleccionando desde el grafo original.
- **Función solución:** Se encuentra un camino que pasa por todas las aristas y no repite ninguna.
- **Función de selección:** Si hay solo una arista se escoge esa. En caso contrario, se escoge cualquier arista que permita que el grafo siga siendo conexo.
- **Criterio de factibilidad:** Sigue habiendo aristas sin explorar.
- **F. Objetivo:** Minimizar el número de pasos por cada arista antes de pasar por todas las aristas del grafo.

Diseño del algoritmo Greedy

ALGORITMO T = Greedy (grafo G=(V,A))

T = Ø // Solución a crear
Si (|V| == 1), **hacer:**
    Devolver T
Fin-Si
N = Número de aristas en A
v = nodo cualquiera de V
Mientras (|T| != N), **hacer:**
    Buscar en A todas las aristas que unen **v** con otros nodos
    Si solo existe una arista **a=(b,v)** donde **b** es un nodo unido a **v** mediante **a**, **hacer:**
        T = T U {a}
        A = A \ {a}
        v = b
    En caso contrario, hacer:
        Escoger una arista **a=(b,v)** donde **b** es un nodo unido a **v** mediante **a** y que mantiene el grafo conexo
        T = T U {a}
        A = A \ {a}
        v = b
    Fin-Si
Fin-Mientras
Devolver T

IMPLEMENTACIÓN Y EFICIENCIA DEL ALGORITMO

```
list<arista> Greedy (const Grafo & g) { O(n² · a²)
    list<arista> solucion;
    list<arista> aristas = g.getAristas();

    if (g.NumNodos() <= 1) {
        return solucion;
    }

    nodo v = g.getNodo(0);

    while (solucion.size() != g.NumAristas()) {
        list<arista> aristasV = aristasUnidasaNodo(v, aristas); O(a)

        if (aristasV.size() == 1) {

            aristas.remove(aristasV.front()); O(n)

            if (aristasV.front().first.id_nodo == v.id_nodo){
                v = aristasV.front().second;
            }
            else {
                v = aristasV.front().first;
                nodo tmp = aristasV.front().first.id_nodo;
                aristasV.front().first = aristasV.front().second;
                aristasV.front().second = tmp;
            }

            solucion.push_back(aristasV.front());

        } else {

            while (!grafoSigueConexo(g, aristasV.back())) {
                aristasV.pop_back();
            }

            aristas.remove(aristasV.back()); O(n)

            if (aristasV.back().first.id_nodo == v.id_nodo){
                v = aristasV.back().second;
            }
            else {
                v = aristasV.back().first;
                nodo aux = aristasV.back().first.id_nodo;
                aristasV.back().first = aristasV.back().second;
                aristasV.back().second = aux;
            }

            solucion.push_back(aristasV.back());
        }
    }

    return solucion;
}
```

O(n² · a²)

O(n² · a)

```
list<arista> aristasUnidasaNodo (nodo &n, list<arista> &aristas) { O(a)
    list<arista> solucion;

    for (arista i : aristas) { nº veces = nº aristas → O(a)
        if (i.first.id_nodo == n.id_nodo || i.second.id_nodo == n.id_nodo) {
            solucion.push_back(i);
        }
    }

    return solucion;
}
```

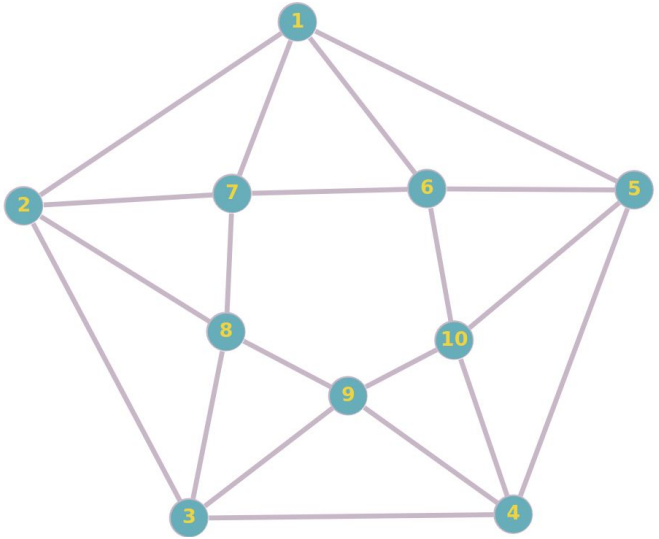
```
bool grafoSigueConexo (const Grafo &g, arista &arist){  O(n² · a)
    Grafo aux;
    aux.setNodos(g.getNodos());
    aux.setAristas(g.getAristas());

    aux.getAristas().remove(arist); O(n)
    aux.BFS(0); O(n² · a)

    for (nodo &n : aux.getNodos()){ O(n)
        if (n.color != NEGRO) return false;
    }

    return true;
}
```

IMPLEMENTACIÓN Y EFICIENCIA DEL ALGORITMO

```
void BFS (int i){ O(n²·a)
    nodo &s = getNodo(i);

    for (nodo &n : nodos){ nº veces = n
        n.color = BLANCO;
    }                                          O(n)

    s.color = GRIS;

    list<nodo*> cola;
    cola.push_back(&s);
    while (!cola.empty()){
        nodo &u = (*cola.front());
        cola.pop_front();

        for (nodo &v : nodos){
            if (nodoRelacionado(v, u)) O(a)
                if (v.color == BLANCO){
                    v.color = GRIS;
                    cola.push_back(&v);
                }
        }                                      O(n·a)

        u.color = NEGRO;                                  O(n²·a)
    }
}
```

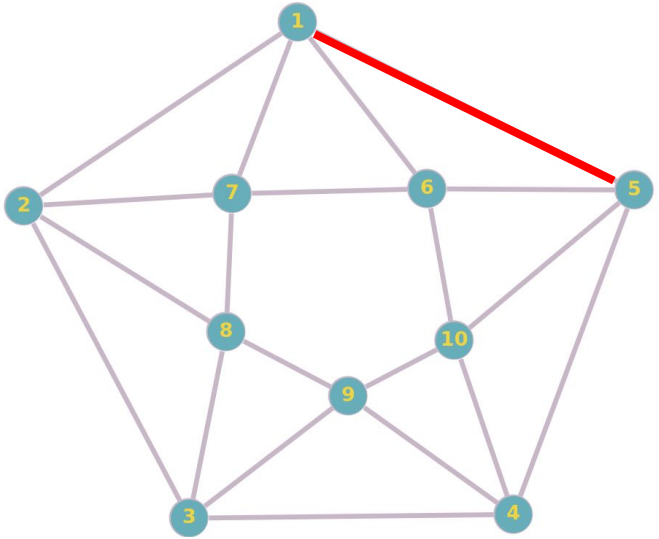
```
bool nodoRelacionado (const nodo &n1, const nodo &n2){ O(a)
    for (arista a : aristas){ nº veces = a → O(a)
        if (a.first.id_nodo == n1.id_nodo && a.second.id_nodo == n2.id_nodo)
            return true;
        else if (a.first.id_nodo == n2.id_nodo && a.second.id_nodo == n1.id_nodo)
            return true;
    }
    return false;
}
```

Ejecuciones - Grafo 1



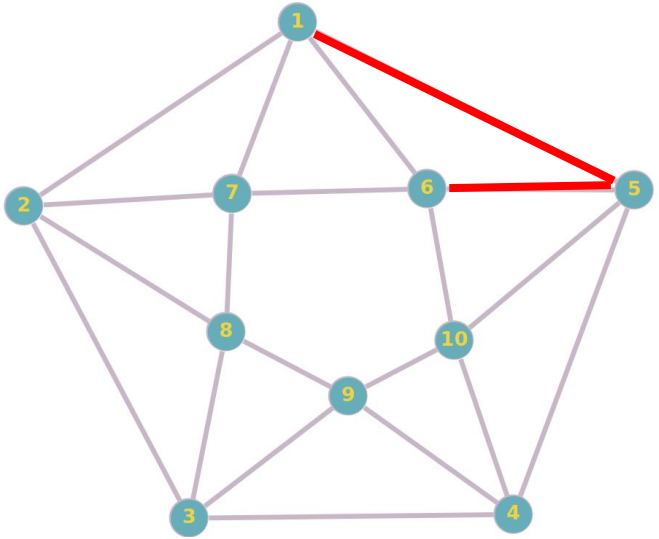

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



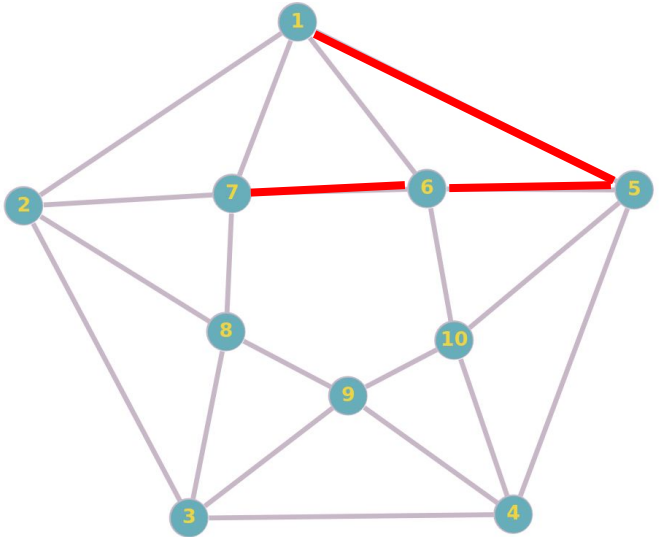

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



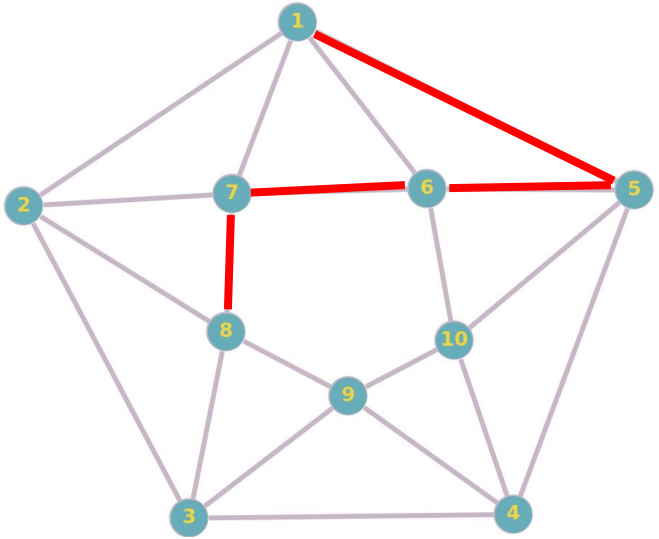

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



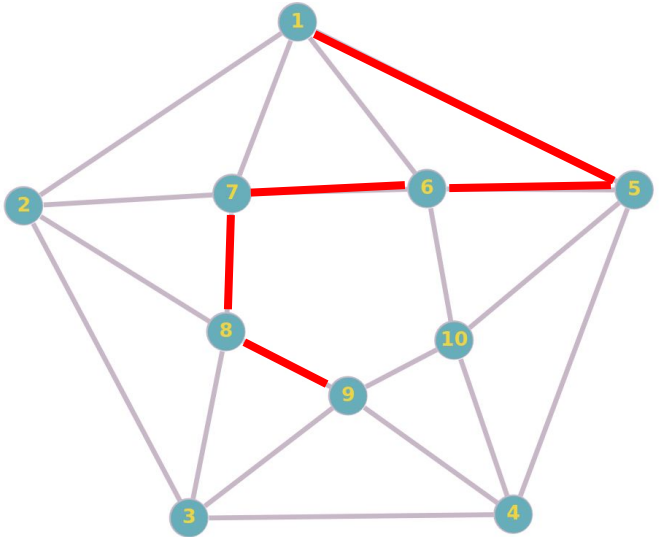

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



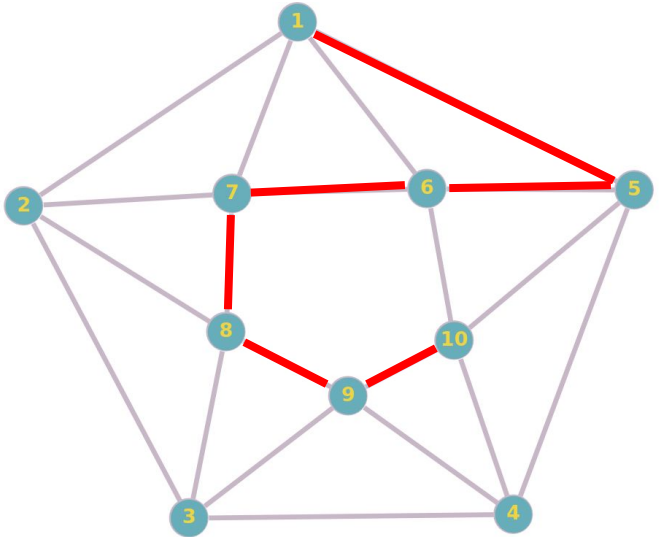

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



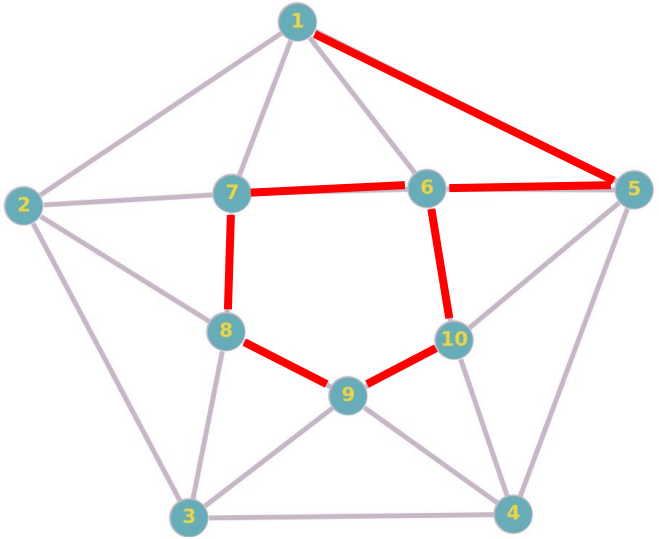

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



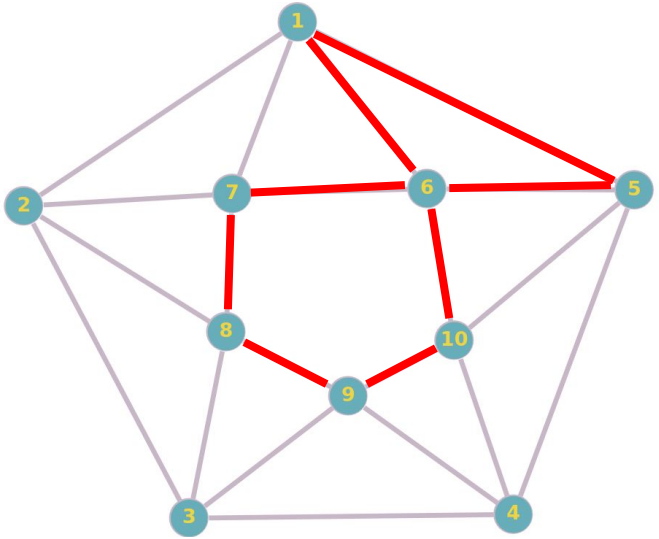

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



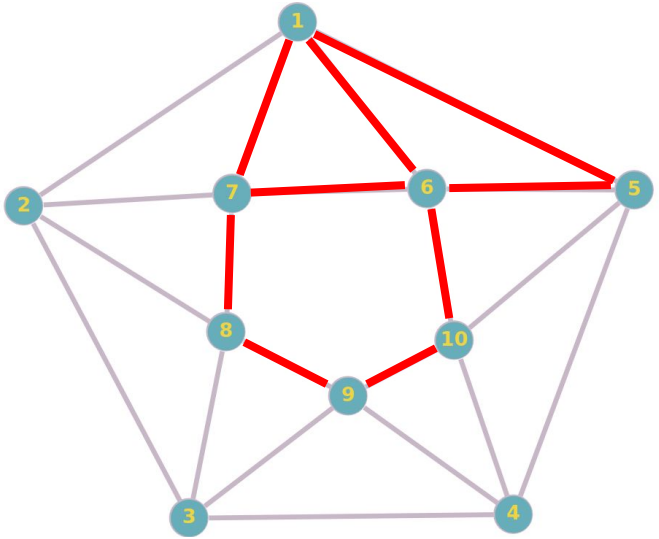

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



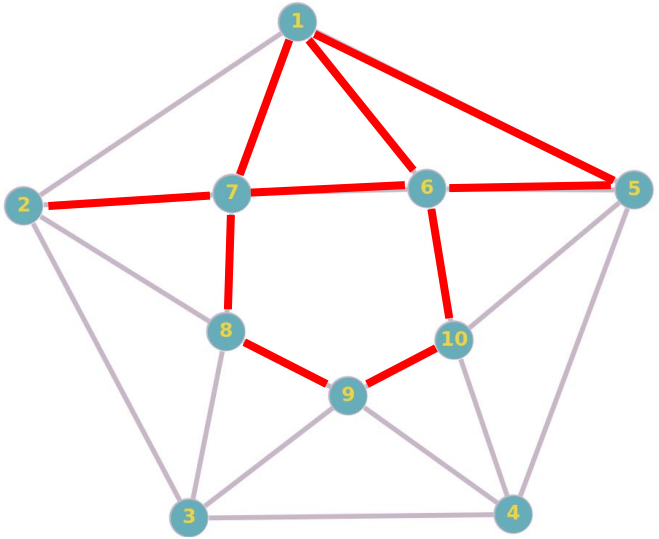

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



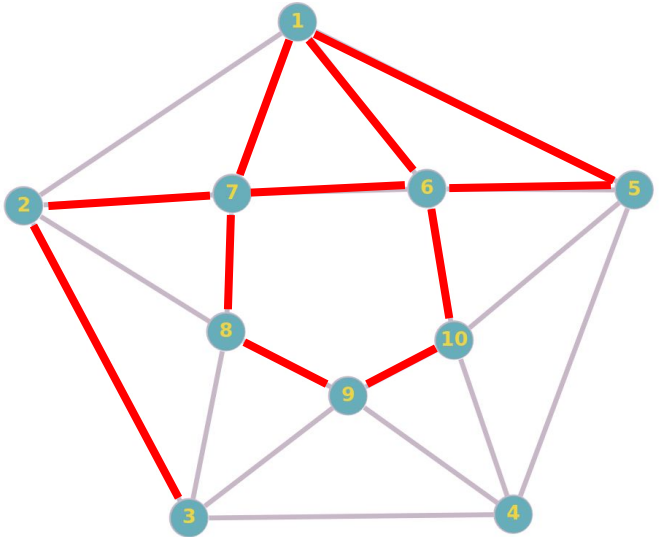

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



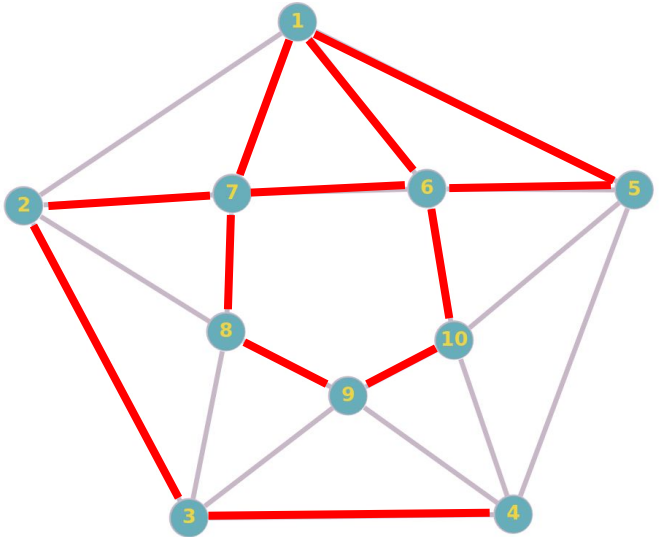

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



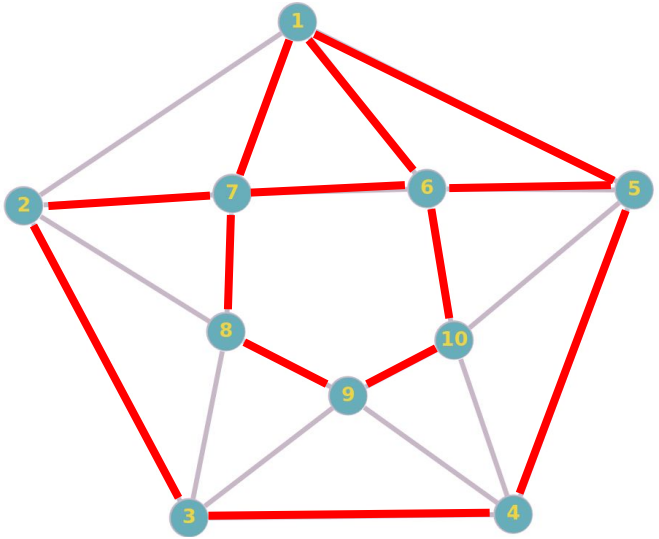

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



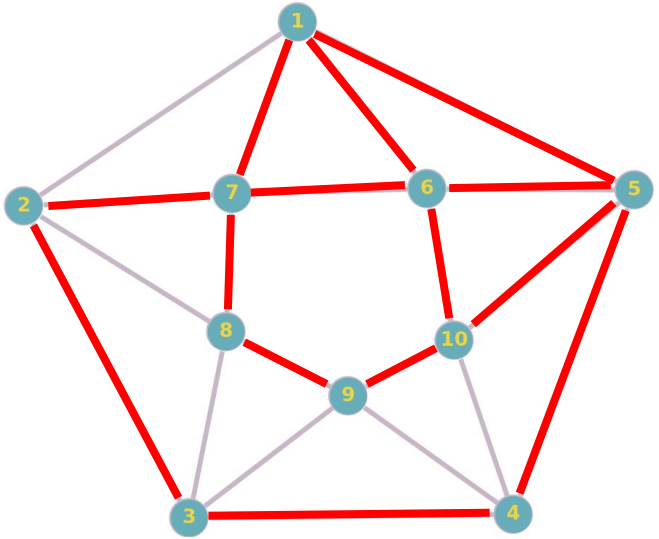

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



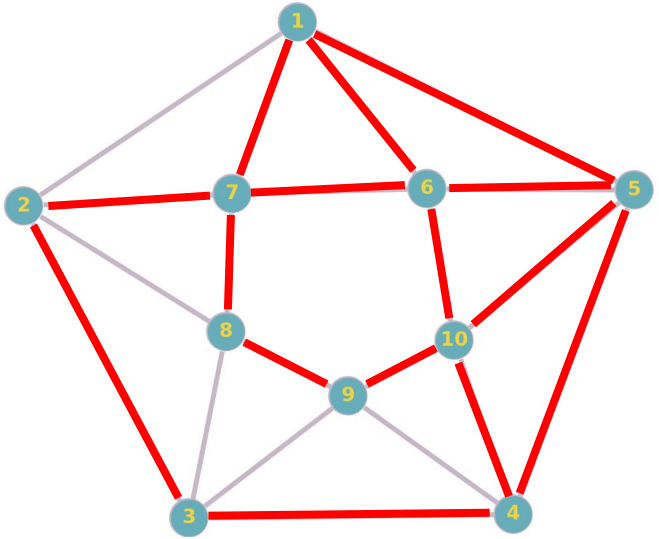

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



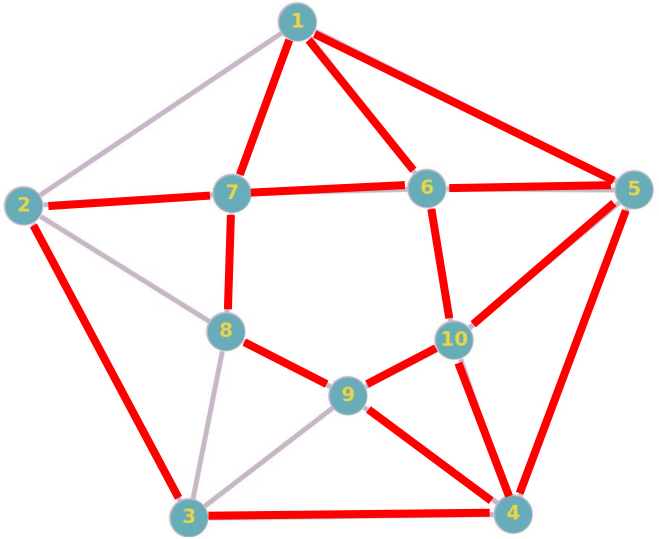

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



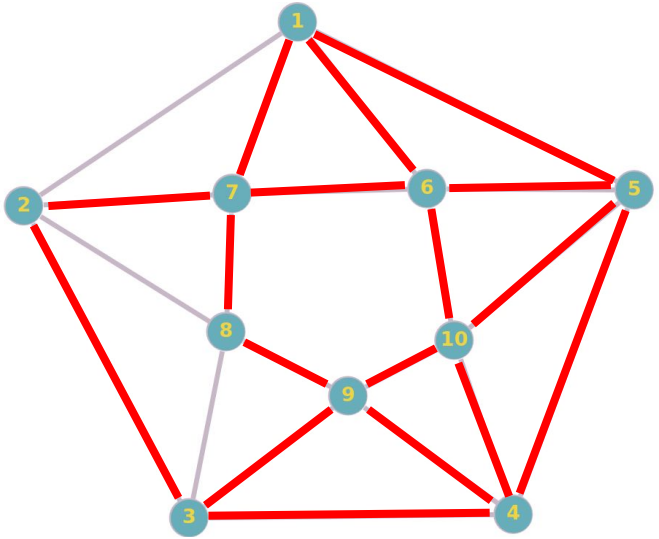

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



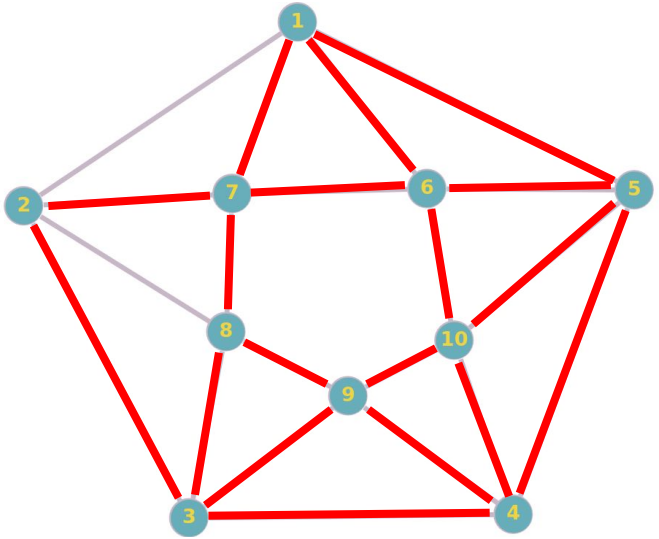

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



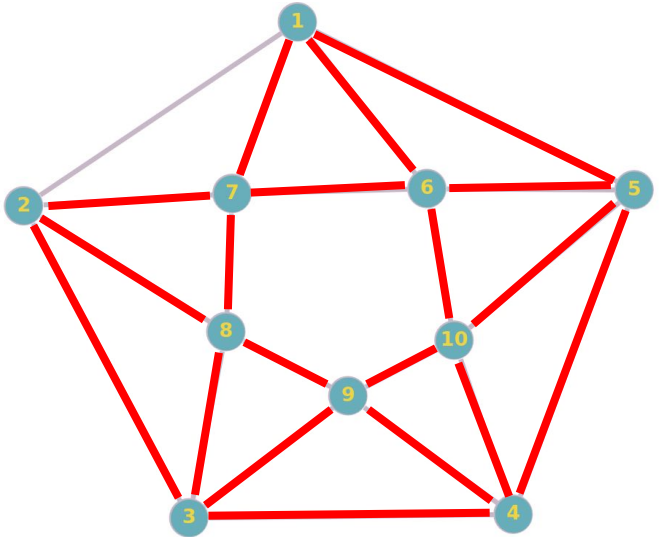

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1



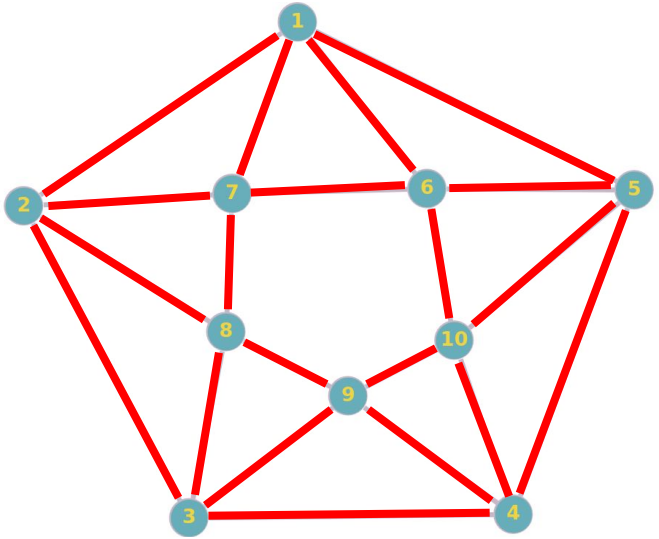

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 1




```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

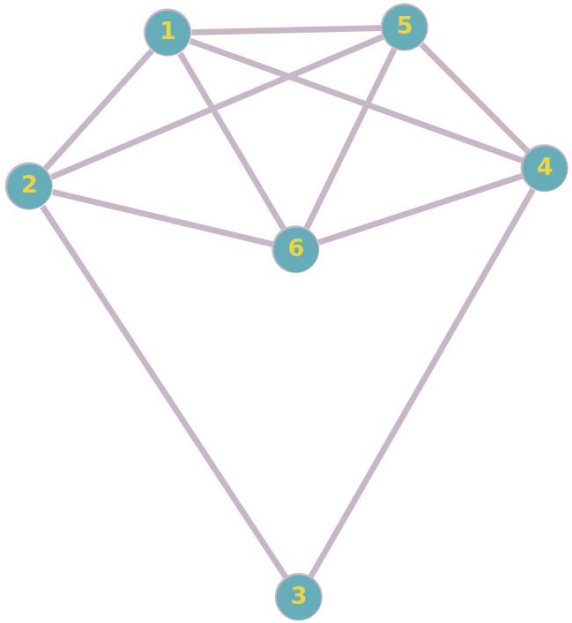
Ejecuciones - Grafo 1



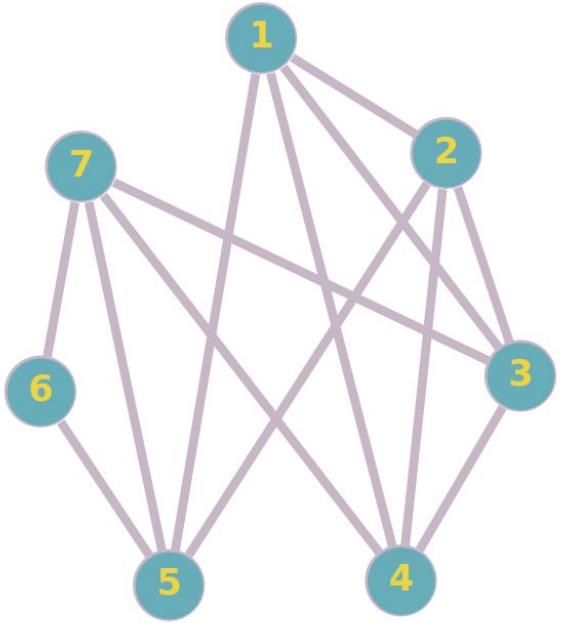

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo1.txt
1 5
5 6
6 7
7 8
8 9
9 10
10 6
6 1
1 7
7 2
2 3
3 4
4 5
5 10
10 4
4 9
9 3
3 8
8 2
2 1
```

Ejecuciones - Grafo 2

```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo2.txt
1 6
6 5
5 4
4 6
6 2
2 5
5 1
1 4
4 3
3 2
2 1
```



Ejecuciones - Grafo 3




```
hossam@DESKTOP-FOF3I96:~/ALG/Practicas-ALG/Practica3$ practica3 grafo3.txt
1 5
5 7
7 3
3 4
4 7
7 6
6 5
5 2
2 4
4 1
1 3
3 2
2 1
```