



Comparación de Algoritmos

Implementa los algoritmos de Ford-Fulkerson y Edmonds-Karp para resolver el problema de máximo flujo en un grafo dirigido.

Ford - Fulkerson

Ford-Fulkerson es un enfoque general para resolver el problema de **flujo máximo** en un grafo dirigido. La idea es encontrar caminos desde el nodo fuente (**s**) al nodo sumidero (**t**) donde aún quede capacidad disponible, aumentar el flujo en ese camino y repetir el proceso hasta que ya no sea posible encontrar más caminos aumentantes.

Seudocódigo:

1. Inicializar flujo = 0 para todas las aristas
2. Mientras exista un camino **s** hasta **t**:
 - 2.1. Encontrar el camino usando una búsqueda DFS
 - 2.2. Determinar la capacidad residual mínima a lo largo del camino
 - 2.3. Para cada arista en el camino **p**:
 - 2.3.1. Incrementar el flujo en la capacidad residual
 - 2.3.2. Decrementar el flujo en la capacidad residual de la arista inversa
3. Devolver el flujo total acumulado en las aristas que salen de la fuente **s**

Edmonds - Karp

Edmonds-Karp es una versión más específica de Ford-Fulkerson que usa **BFS (búsqueda en anchura)** para encontrar caminos aumentantes en lugar de DFS. Esto hace que el algoritmo sea más predecible y mejora su eficiencia en algunos casos.

Seudocódigo:

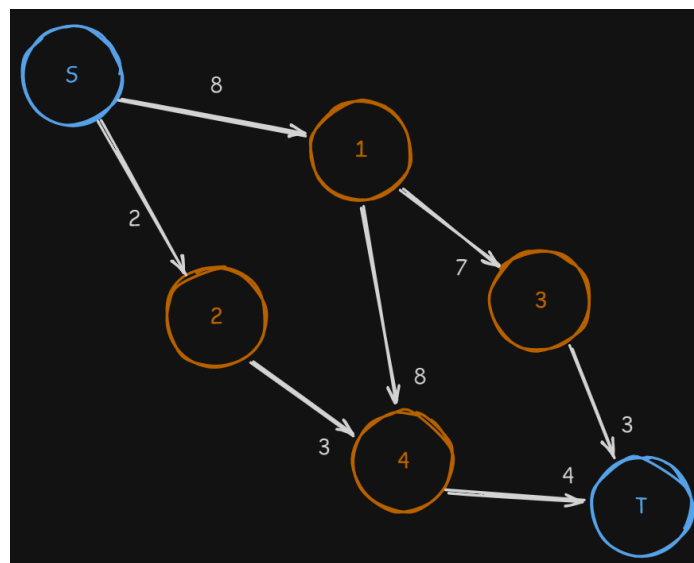
1. Inicializar flujo = 0 para todas las aristas
2. Mientras exista un camino s hasta t:
 - 2.1. Encontrar el camino usando una búsqueda BFS (tramo mas corto)
 - 2.2. Determinar la capacidad residual mínima a lo largo del camino
 - 2.3. Para cada arista en el camino p:
 - 2.3.1. Incrementar el flujo en la capacidad residual
 - 2.3.2. Decrementar el flujo en la capacidad residual de la arista inversa
3. Devolver el flujo total acumulado en las aristas que salen de la fuente s

Pruebas con diferentes grafos:

Genera grafos con las siguiente configuraciones:

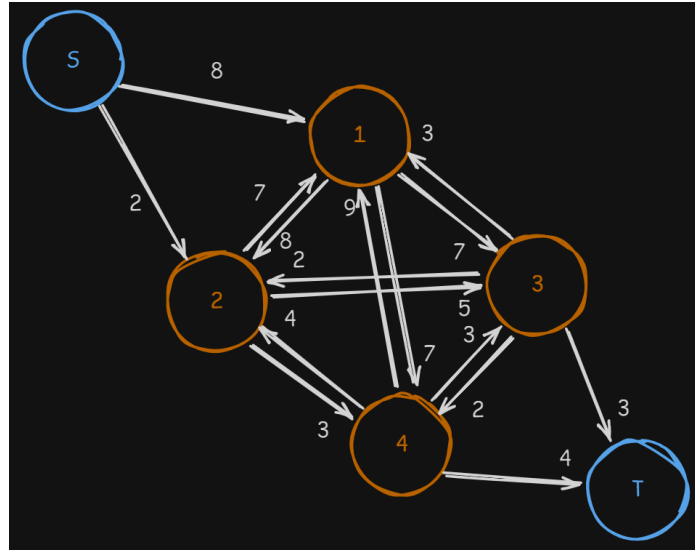
- **Pequeños y Escasos:** Grafos con pocas aristas (baja densidad) y pocos nodos.

Un grafico para ver como seria un grafo pequeño y escaso, pero en las pruebas se uso un grafo de 5 nodos y 5 aristas.

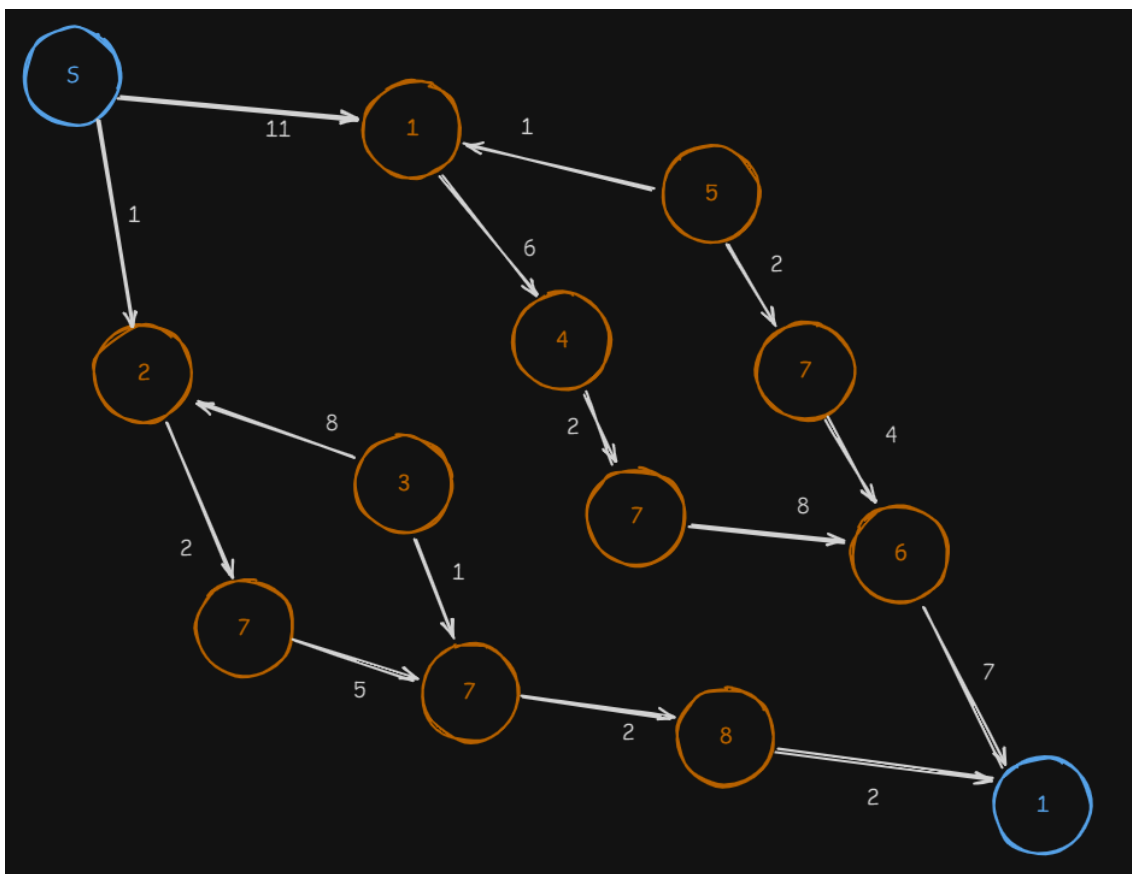


- **Pequeños y Densos:** Grafos con muchas aristas (alta densidad) y pocos nodos.

Un grafico para ver como seria un grafo pequeño y denso, pero en las pruebas se uso un grafo de 10 nodos y 199 aristas.

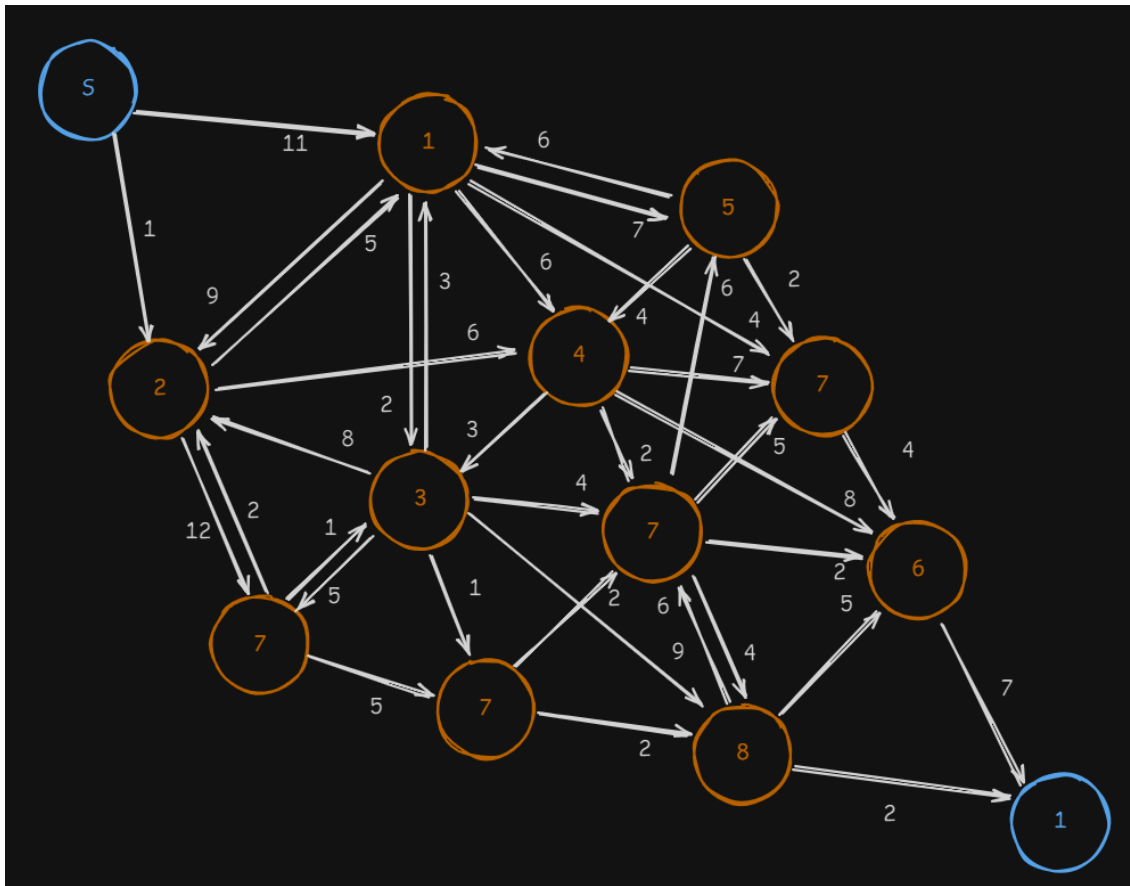


- **Grandes y Escasos:** Grafos con muchos nodos pero pocas aristas.
Un grafico para ver como seria un grafo grande y escaso, pero en las pruebas se uso un grafo de 122 nodos y 199 aristas.



- **Grandes y Densos:** Grafos con muchos nodos y muchas aristas.
Un grafico para ver como seria un grafo grande y denso, pero en las pruebas se uso un grafo de

442 nodos y 998 aristas.



Comparación Empírica

Ejecuta ambos algoritmos en cada uno de los grafos generados. Registra:

- El tiempo de ejecución de cada algoritmo.
- El número de iteraciones (o búsquedas) realizadas por cada algoritmo.
- El flujo máximo obtenido (ambos algoritmos deben dar el mismo resultado).

Grafos:

- Pequeños y escasos

FORD FULKERSON

```
1 Tiempo de ejecución: 0.00035 segundos
2 Iteraciones: 3
3 Max Flow: 3
4
```

EDMONDS KARP

```
1 Tiempo de ejecución: 0.000509 segundos
2 Iteraciones: 2
3 Max Flow: 3
4
```

- Pequeños y densos

FORD FULKERSON

```
1 Tiempo de ejecución: 0.024727 segundos
2 Iteraciones: 4000
3 Max Flow: 4000
4
```

EDMONDS KARP

```
1 Tiempo de ejecución: 0.000811 segundos
2 Iteraciones: 4
3 Max Flow: 4000
4
```

- Grandes y escasos

FORD FULKERSON

```
1 Tiempo de ejecución: 0.000918 segundos
2 Iteraciones: 2
3 Max Flow: 2
4
```

EDMONDS KARP

```
1 Tiempo de ejecución: 0.000799 segundos
2 Iteraciones: 1
3 Max Flow: 2
4
```

- Grandes y densos

FORD FULKERSON

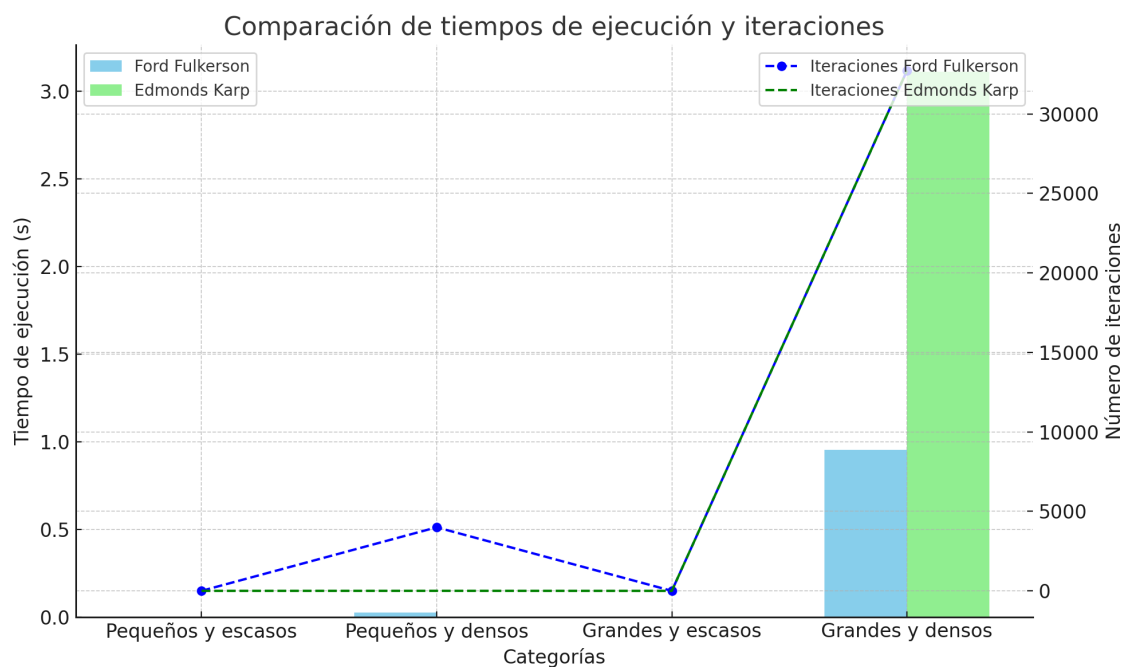
```
1 Tiempo de ejecución: 0.951697 segundos
2 Iteraciones: 32724
3 Max Flow: 32724
4
```

EDMONDS KARP

```
1 Tiempo de ejecución: 3.1102 segundos
2 Iteraciones: 32724
3 Max Flow: 32724
4
```

Gráfico

Gráficos o tablas que comparen el rendimiento de ambos algoritmos en función del tiempo de ejecución y el número de iteraciones.



En el gráfico:

- Las barras representan los tiempos de ejecución en segundos.
- Las líneas discontinuas representan el número de iteraciones.

Conclusiones

Ventajas

- Ford Fulkerson
 - Es fácil de implementar con DFS.
 - Puede encontrar soluciones en casos donde los flujos son números enteros.

- Edmonds Karp
 - Garantiza que el número de iteraciones esté acotado, lo que da una mejor complejidad que Ford-Fulkerson con DFS.
 - Siempre encuentra el camino más corto en términos de aristas, lo que puede ser ventajoso en grafos densos.

Desventajas

- Ford Fulkerson
 - Con DFS puede ser muy lento en ciertos grafos, especialmente si no se prioriza el camino con mayor capacidad residual.
- Edmonds Karp
 - Aunque es más eficiente que Ford-Fulkerson con DFS, puede seguir siendo lento en algunos grafos grandes, ya que la complejidad es cuadrática en el número de aristas.