

O PROBLEMA DO FLUXO MÁXIMO

Universidade Federal de Sergipe

Programa de Pós Graduação em Ciência da Computação

Projeto e Análise de Algoritmos

Prof. Dr. Leonardo Nogueira Matos

Discentes:

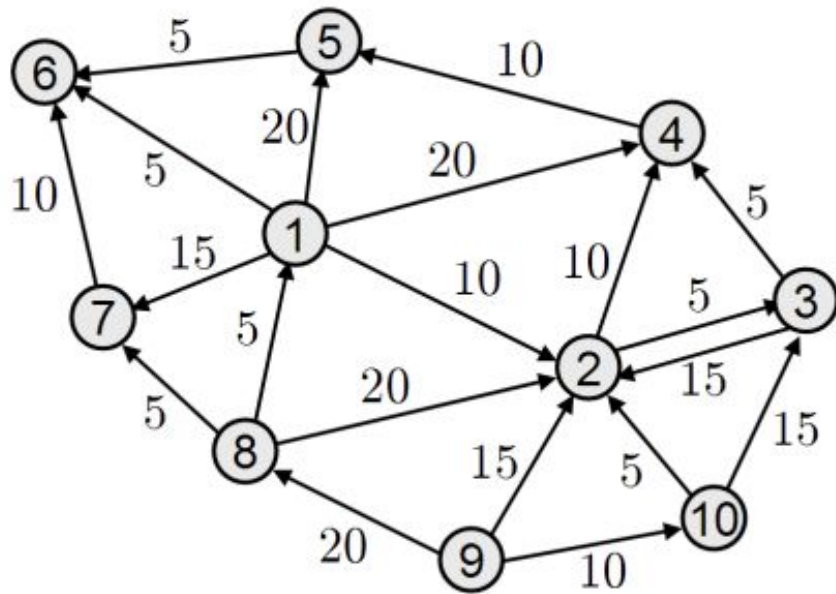
Luiz Carlos da Conceição Júnior

Márcio da Silva Alves

ÍNDICE

- Introdução;
- O algoritmo de Ford-Fulkerson;
- Código-fonte;
- Conclusão.

INTRODUÇÃO



O problema do fluxo máximo consiste em encontrar o maior fluxo possível de um nó de origem até um nó de destino em um grafo direcionado, considerando que cada aresta tem uma capacidade, ou seja, o máximo que pode passar por ela.

INTRODUÇÃO

- Imagine uma rede de computadores onde cada cabo suporta no máximo 100 mbps.
- Qual seria a melhor forma de transmitir todos os dados de forma eficiente, utilizando todos os caminhos possíveis, sem ultrapassar a capacidade?

O ALGORITMO DE FORD-FULKERSON

- O algoritmo de Ford-Fulkerson é o método mais comum de se resolver o problema do fluxo máximo.
- Cada iteração começa com um fluxo f que respeita a capacidade das arestas.
- A primeira iteração começa com o fluxo nulo.
- O processo iterativo consiste no seguinte: enquanto existe pseudocaminho aumentador,
 1. encontre um pseudocaminho aumentador P ,
 2. calcule a capacidade residual δ de P ,
 3. envie δ unidades de fluxo ao longo de P (e atualize f).

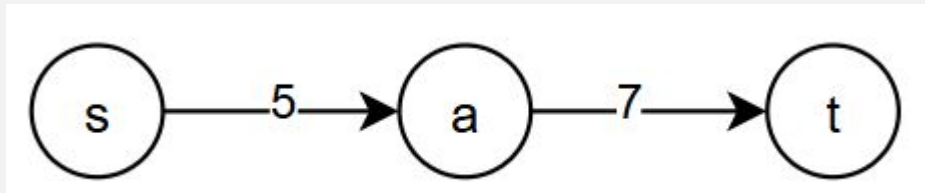
O ALGORITMO DE FORD-FULKERSON

Conceitos:

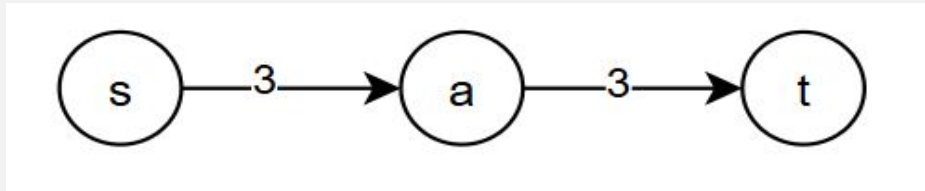
- Pseudocaminho:
 - Sequência de vértices onde, para cada par v w , ou $v-w$ é um arco, ou $w-v$ é um arco.
- Pseudocaminho aumentante:
 - Vai do vértice inicial ao final do grafo;
 - Nenhum arco direto está cheio;
 - Nenhum arco reverso está vazio.

O ALGORITMO DE FORD-FULKERSON

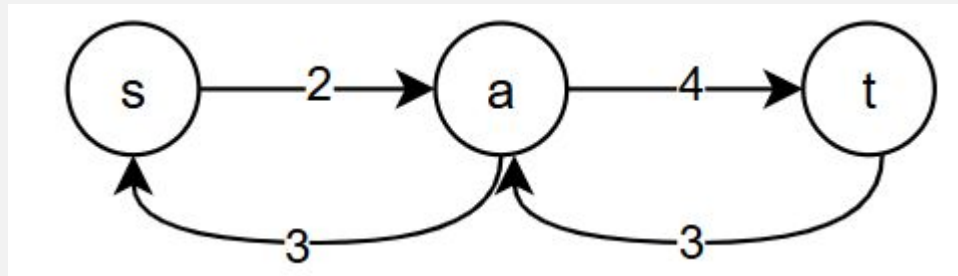
- Grafo capacitado G



- Grafo de fluxo G^F



- Grafo residual G^R



O ALGORITMO DE FORD-FULKERSON

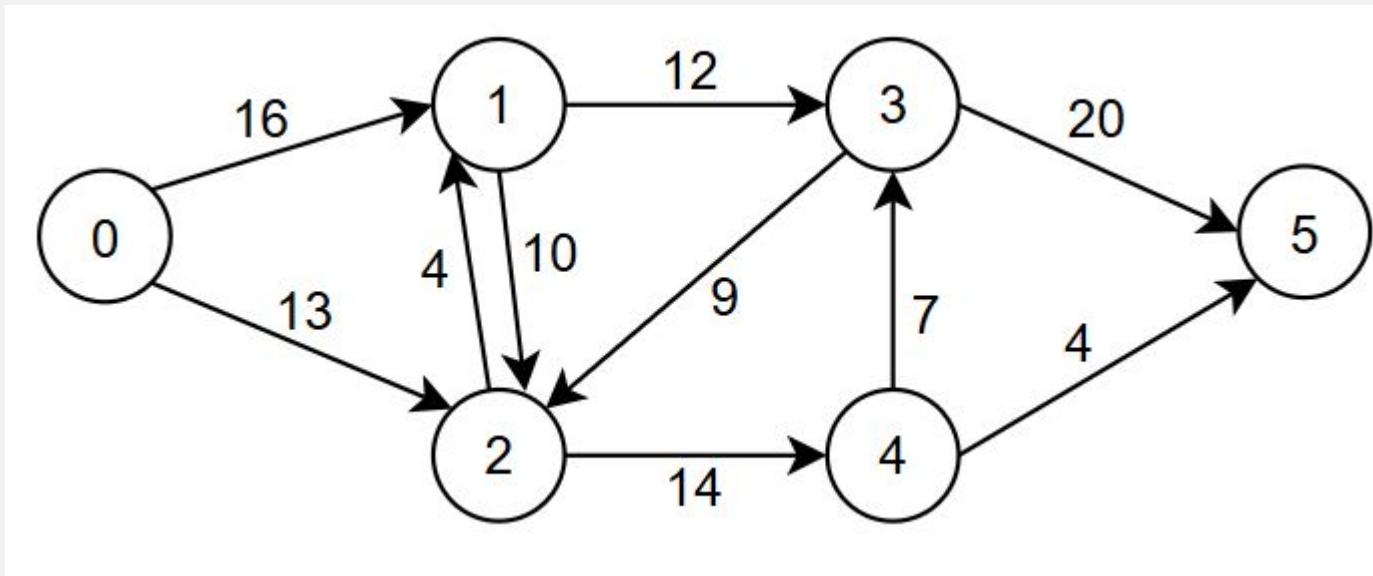
- A aplicação prática do algoritmo é o Edmonds-Karp, que utiliza busca em largura para definir o caminho aumentador.
- Enquanto esse caminho existir, descobre-se seu fluxo, atualiza as capacidades restantes das arestas, e soma esse fluxo a uma variável que guardará o fluxo total.
- No fim, retorna-se essa variável, que é o fluxo máximo.

O ALGORITMO DE FORD-FULKERSON

- Complexidade de Ford-Fulkerson e Edmonds-Karp
- Ford-Fulkerson : $O(M \cdot |E|)$ número de iterações vezes o número de arestas
- Edmonds-Karp : $O(|E|^2 |V|)$ quadrado do número de arestas vezes o número de vértices

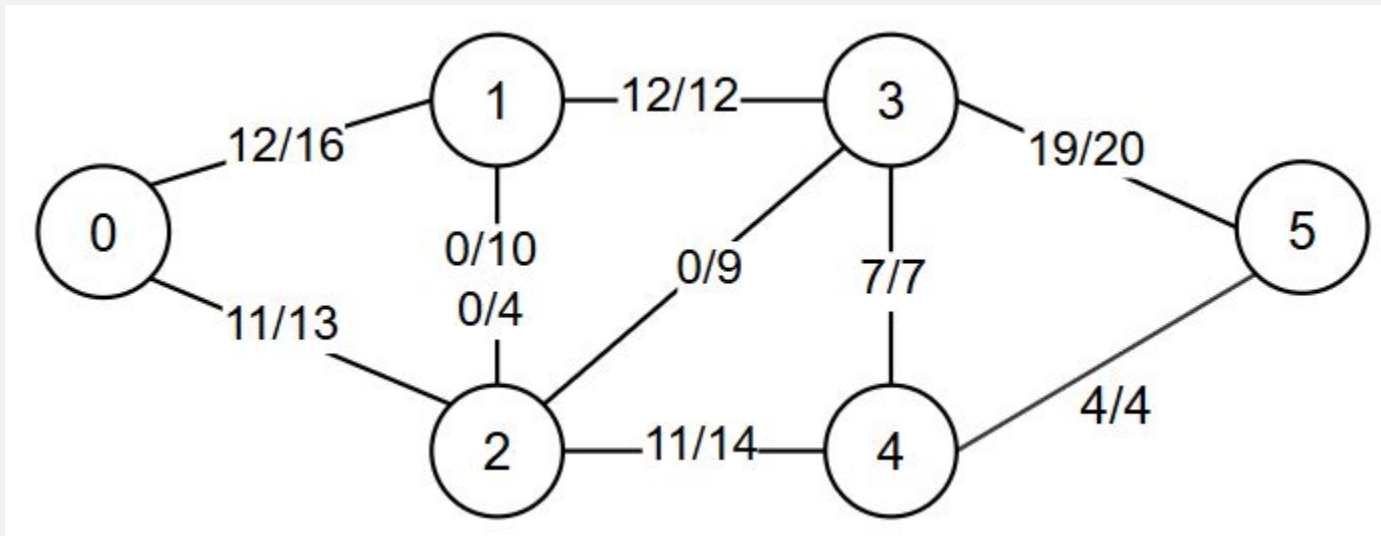
O ALGORITMO DE FORD-FULKERSON

Grafo do código fonte abaixo:



O ALGORITMO DE FORD-FULKERSON

Grafo do código fonte abaixo:



CÓDIGO-FONTE

- Busca em largura:

```
from collections import deque

def bfs(residual, origem, destino, caminho):
    visitado = [False] * len(residual)
    fila = deque([origem])
    visitado[origem] = True

    while fila:
        u = fila.popleft()
        for v, capacidade in enumerate(residual[u]):
            if not visitado[v] and capacidade > 0:
                visitado[v] = True
                caminho[v] = u
                if v == destino:
                    return True
                fila.append(v)
    return False
```

CÓDIGO-FONTE

- Edmonds-Karp:

```
def edmonds_karp(grafo, origem, destino):
    n = len(grafo)
    residual = [linha[:] for linha in grafo]
    caminho = [-1] * n
    fluxo_maximo = 0

    while bfs(residual, origem, destino, caminho):
        fluxo = float("inf")
        v = destino
        while v != origem:
            u = caminho[v]
            fluxo = min(fluxo, residual[u][v])
            v = caminho[v]

        v = destino
        while v != origem:
            u = caminho[v]
            residual[u][v] -= fluxo
            residual[v][u] += fluxo
            v = caminho[v]

        fluxo_maximo += fluxo

    return fluxo_maximo
```

CÓDIGO-FONTE

- Função principal:

```
grafo = [  
    [0, 16, 13, 0, 0, 0],  
    [0, 0, 10, 12, 0, 0],  
    [0, 4, 0, 0, 14, 0],  
    [0, 0, 9, 0, 0, 20],  
    [0, 0, 0, 7, 0, 4],  
    [0, 0, 0, 0, 0, 0],  
]  
origem = 0  
destino = 5  
  
print("Fluxo máximo:", edmonds_karp(grafo, origem, destino))
```

LINK DO GIT

- Segue link do git para visualização

<https://github.com/MarcioUfs/UFS-PROCC-PAA-PROBLEMA-DO-FLUXO-MAXIMO>