Fluxo (Máximo) de Custo Mínimo

Augusto Damschi Bernardi

Aula passada

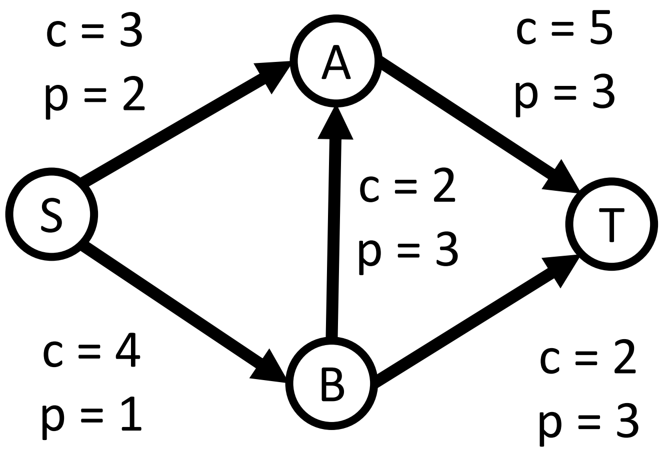
- Redes e Fluxos
- Fluxo Máximo
  - Ford-Fulkerson
  - Edmonds-Karp
  - Dinic
- Fluxo Máximo – Corte Mínimo
- Modelar problemas como fluxo e corte

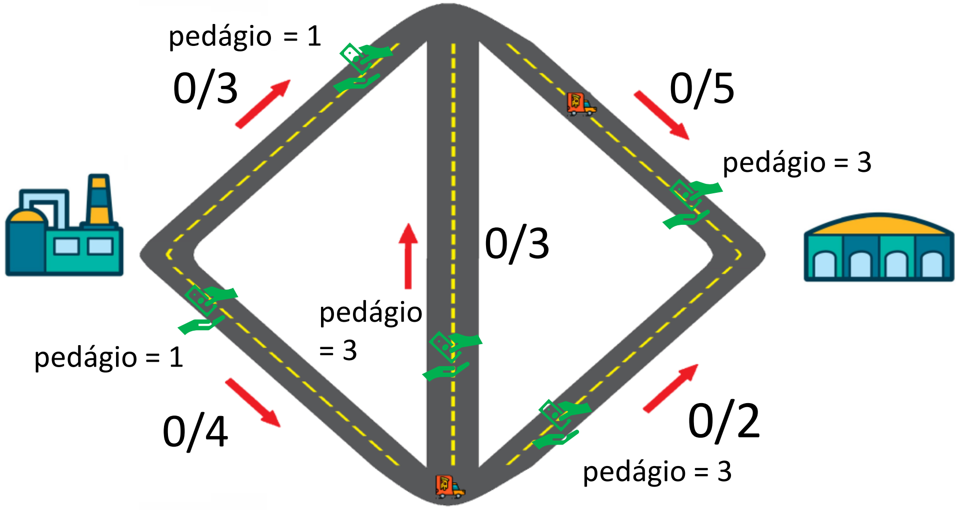
Link da aula passada:

- https://www.youtube.com/watch?v=ApIaUSezpfI
- https://bit.ly/3fS18mz

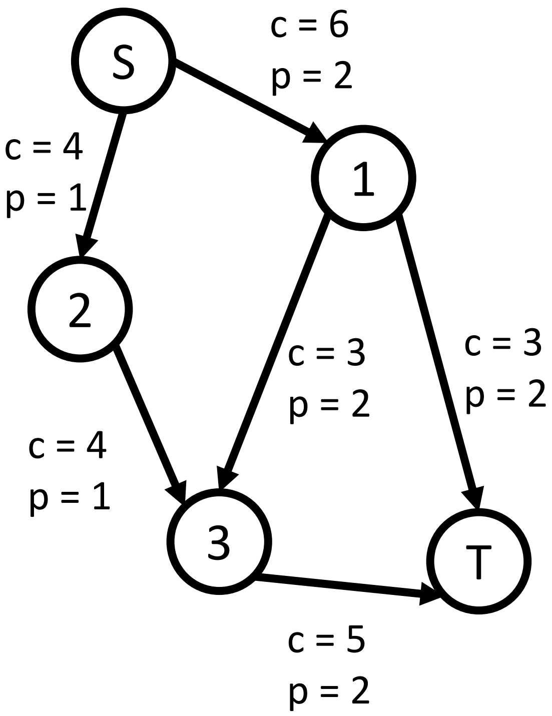
Problema

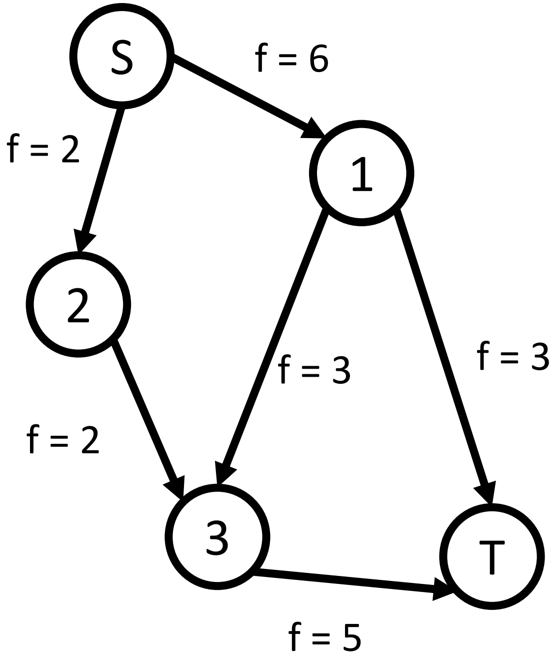
- Rede com custo:
  - Grafo orientado **G = {V, E}**
  - Dois vertices especiais: Fonte **S** e Dreno **T**
  - Cada aresta **e** tem uma capacidade **c(e)**
  - Cada aresta **e** tem um custo **p(e)**
- Analogia: Sistema de estradas de mão única.
  - Fonte: Fábrica de onde saem caminhões
  - Dreno: Depósito onde os caminhões devem chegar
  - Cada estrada com capacidade em caminhões /minuto.
  - O custo de cada aresta é o preço de pedágio por caminhão.
- Custo total de um fluxo **f** é : $\sum_{e \in E} f(e)p(e)$



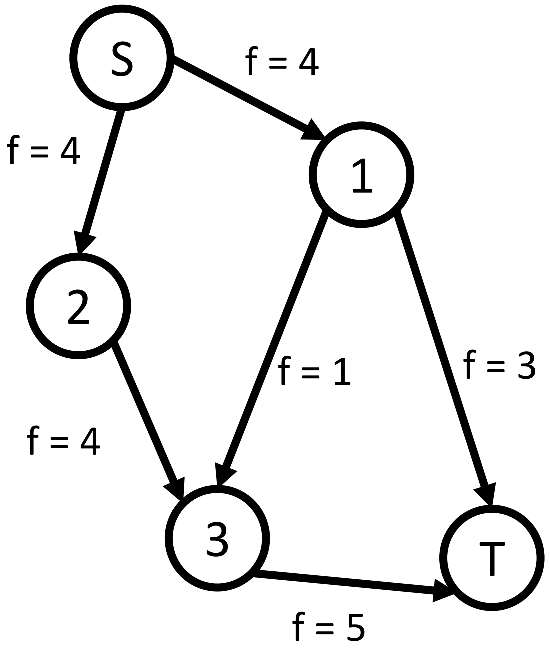


Problema





- Fluxo = 8
- Custo = 6*2 + 2*1 + 3*2 + 2*1 + 3*2 + 5*2 = 38



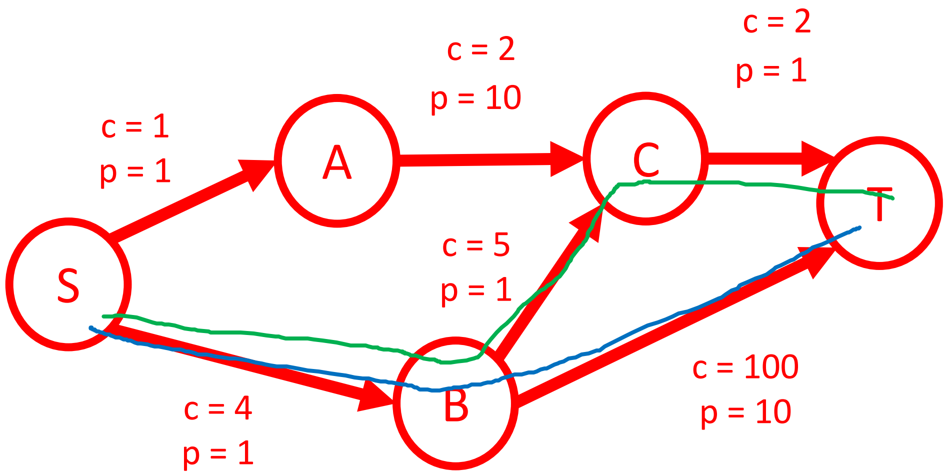
- Fluxo = 8
- Custo = 4*2 + 4*1 + 3*2 + 4*1 + 1*2 + 5*2 = 34

Problema

- Dada a rede com capacidades e custos para as arestas:
    - **Problema do min cost max flow**: Qual o fluxo de valor máximo de menor custo possível?
    - **Problema do min cost flow:** Qual o fluxo de valor K de menor custo possível?

Algoritmo para encontrar fluxo máximo de custo mínimo

- Ideia para fluxo máximo de custo mínimo:
  - Enquanto for existir um caminho no grafo residual, adicionamos fluxo.
  - Mas adicionamos sempre em um caminho de custo mínimo.
- Algoritmo:
  - Inicia fluxo em todas as arestas com zero
  - Cria grafo residual G_R. Para cada aresta **e** em **G**:
    - $c_R(e) = c(e)$ e $p_R(e) = p(e)$
    - $c_R(e') = 0$ e $p_R(e') = -p(e)$
  - Enquanto existir um caminho de **S** até **T** em G_R(sem arestas com $c_R(e) = 0$).
    - Escolhemos um caminho **P** mínimo **usando os custos**.
    - Seja **x** a menor **capacidade** de uma aresta em **P**.
    - Para todo **e** em **P**
      - $c_R(e) \mathrel{-}= x$
      - $c_R(e') \mathrel{+}= x$
      - $f(e) = c(e) - c_R(e)$

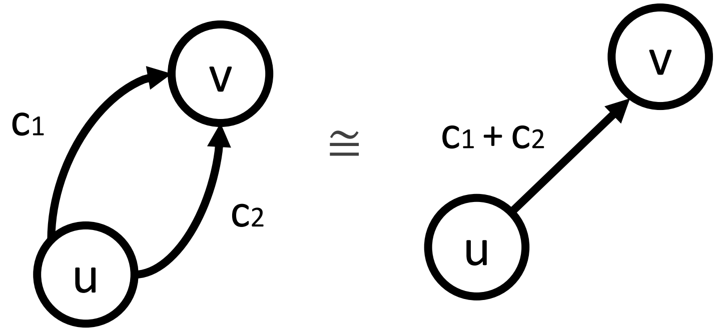
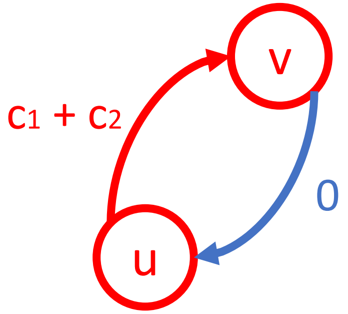
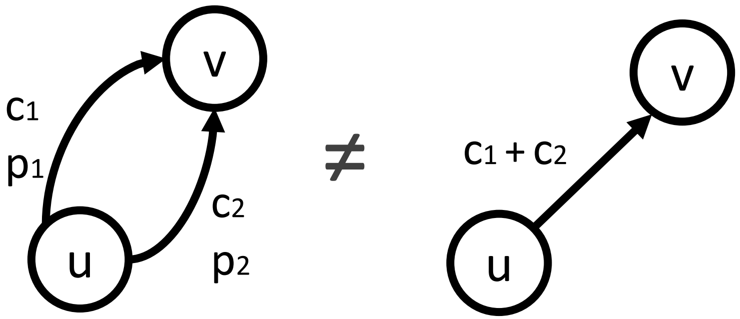
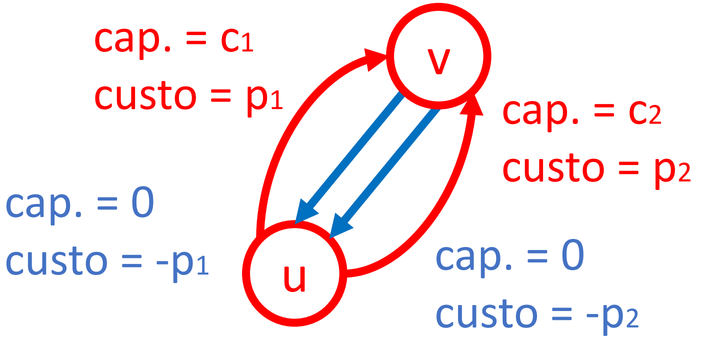


Complexidade:
O(T(E,V) F)

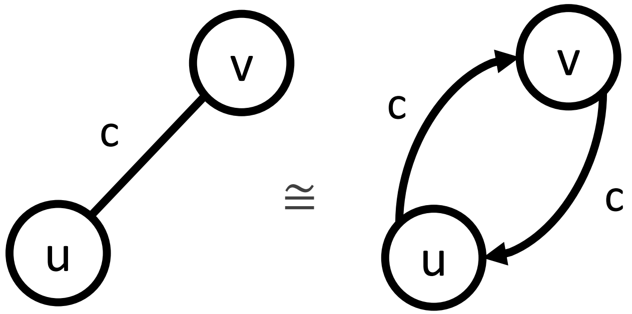
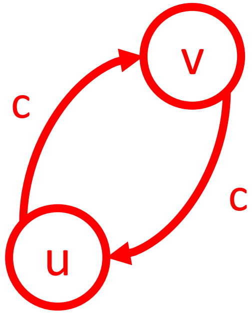
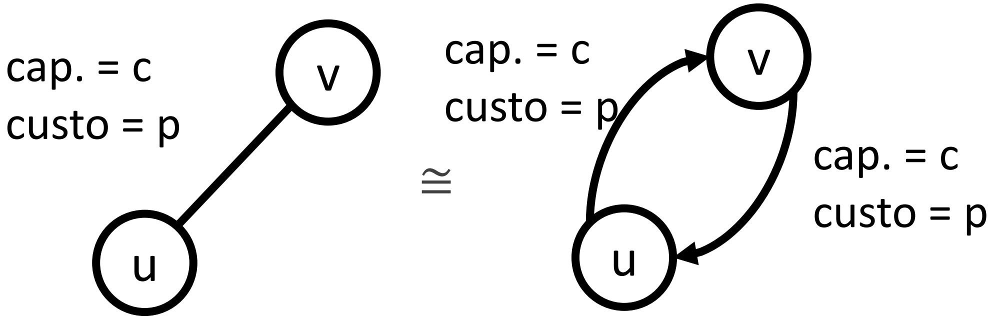
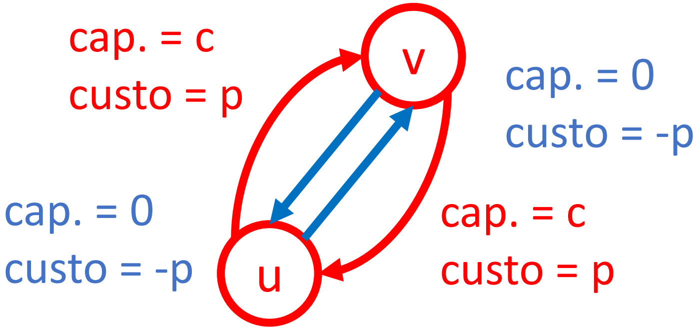
Algoritmo para encontrar fluxo de custo mínimo

- Ideia para fluxo máximo de custo mínimo:
  - Enquanto for existir um caminho no grafo residual, e o fluxo total for menor que o desejado adicionamos fluxo.
  - Mas adicionamos sempre em um caminho de custo mínimo.
- Algoritmo:
  - Inicia fluxo em todas as arestas com zero
  - Cria grafo residual $\mathbf{G_R}$. Para cada aresta **e** em **G**:
    - $\mathbf{c_R(e) = c(e)}$ e $\mathbf{p_R(e) = p(e)}$
    - $\mathbf{c_R(e') = 0}$ e $\mathbf{p_R(e') = -p(e)}$
  - Enquanto existir um caminho de **S** até **T** em $\mathbf{G_R}$(sem arestas com $\mathbf{c_R(e)}$ = 0), e não tivermos o fluxo desejado.
    - Escolhemos um caminho **P** mínimo **usando os custos**.
    - Seja **x** o menor valor entre menor **capacidade** de uma aresta em **P** e **fluxo desejado – fluxo atual**.
    - Para todo **e** em **P**
      - $\mathbf{c_R(e) \mathrel{-}= x}$
      - $\mathbf{c_R(e') \mathrel{+}= x}$
      - $\mathbf{f(e) = c(e) - c_R(e)}$

Observação sobre arestas múltiplas

	Rede	Grafo Residual no início
Fluxo Máximo		
Fluxo (Máximo) de Custo Mínimo		

Observação sobre arestas não orientadas

	Rede	Grafo Residual no início
Fluxo Máximo		
Fluxo (Máximo) de Custo Mínimo		

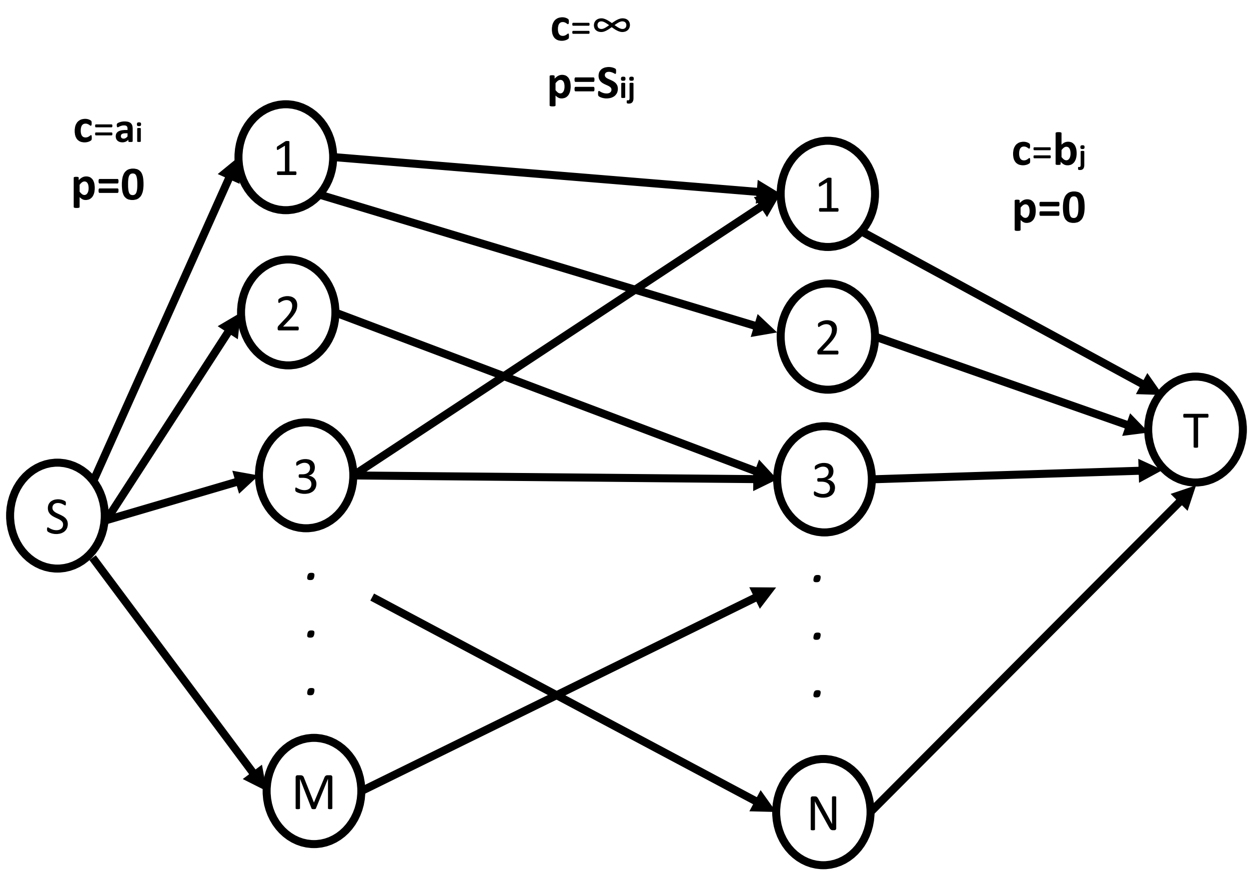
Exemplo 1

SPOJ: ASSIGN4

- Exitem **M** perfis de trabalhadores e **N** tipos de vaga a serem preenchidas.
- Você tem $\mathbf{a_i}$ trabalhadores do tipo **i**, e $\mathbf{b_j}$ vagas do tipo **j**.
- Custa $\mathbf{S_{ij}}$ para contratar um trabalhador do tipo **i** para uma vaga do tipo **j**.
- Qual o menor custo para preencher todas as vagas?

Restrições:

- $1 \leq \mathbf{N}, \mathbf{M} \leq 200$
- $1 \leq \mathbf{a_i}, \mathbf{b_j} \leq 3\ 10^4$
- $1 \leq \mathbf{S_{ij}} \leq 10^4$



Tempo: O(NM + min(∑ a_i, ∑ b_j)) $\cong 6\ 10^6$

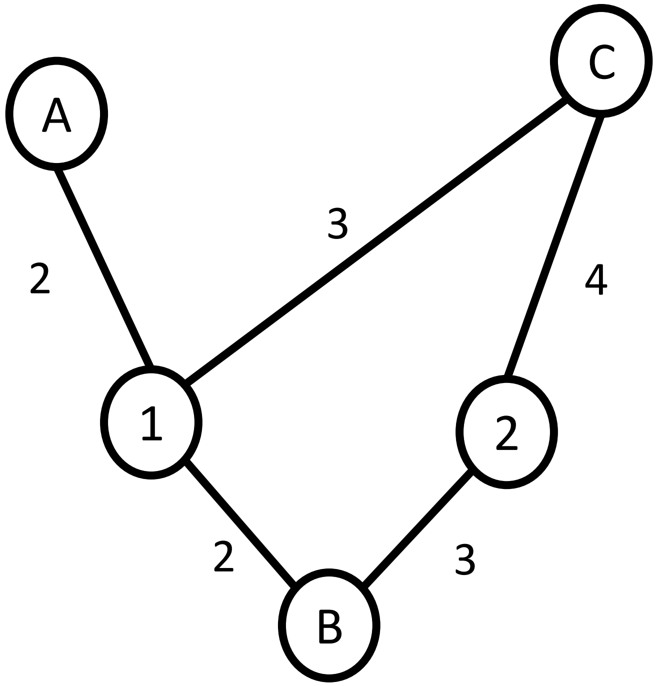
Exemplo 2

- Você tem um mapa de uma cidade como um grafo não-orientado. Cada aresta tem o tamanho da rua $\mathbf{s_i}$.

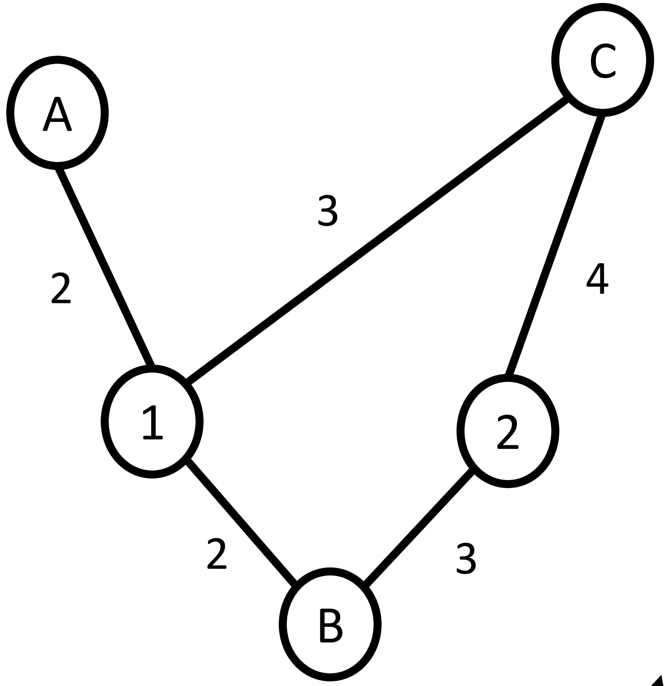
- Você quer ir do vértice **A** até o **B**, e então do **B** até o **C**, sem passar duas vezes por nenhuma rua no trajeto inteiro.

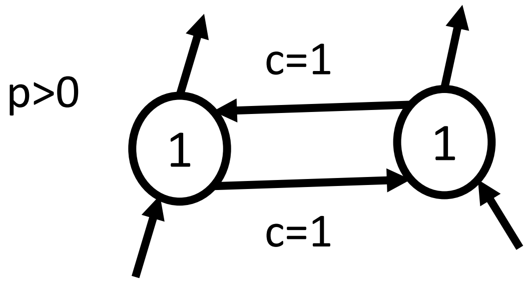
- **N** junções de rua, **M** ruas.

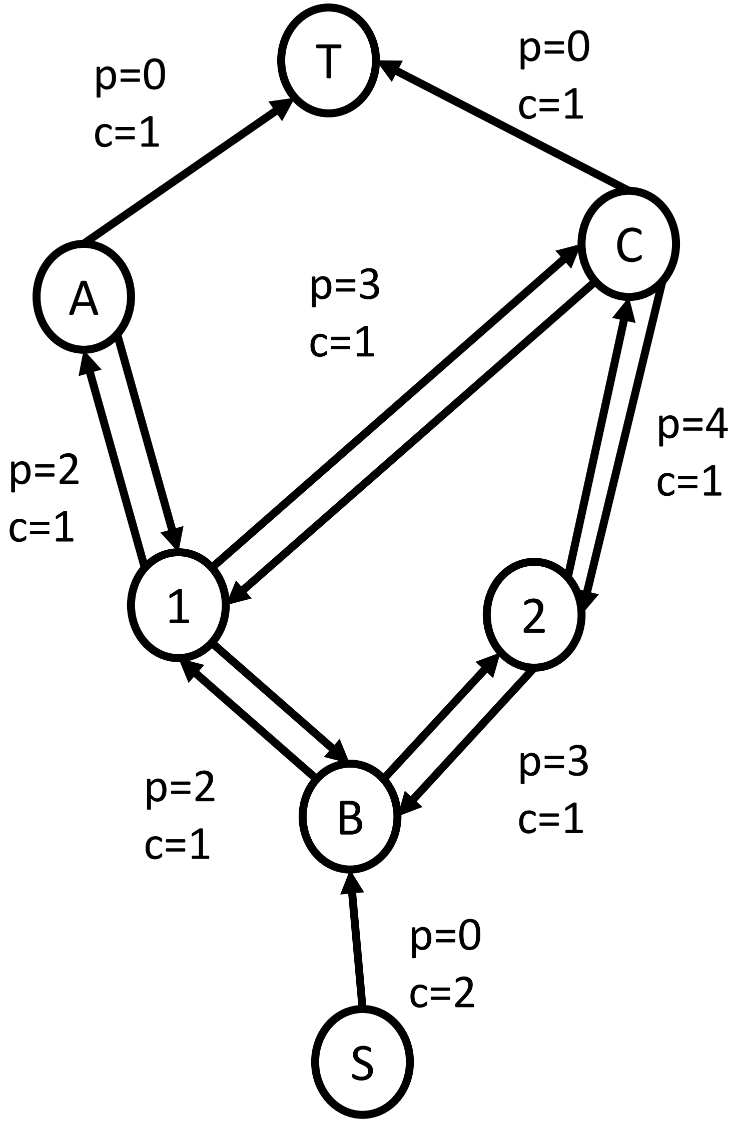
Restrições:
- $1 \leq \mathbf{N}, \mathbf{M} \leq 3\ 10^4$
- $1 \leq \mathbf{s_i} \leq 10^4$



Exemplo 2


A
C
1
2
B
2
3
4
2
3


p>0
c=1
c=1
1
1


T
p=0
c=1
p=0
c=1
A
C
p=3
c=1
p=2
c=1
p=4
c=1
1
2
p=2
c=1
p=3
c=1
B
p=0
c=2
S

Referências

Explicação + Implementação

- **CP Algorithms:** Minimum-cost flow - Successive shortest path algorithm

  https://cp-algorithms.com/graph/min_cost_flow.html

  https://bit.ly/3fV14CJ

- **TopCoder:** Minimum Cost Flow

  https://www.topcoder.com/thrive/articles/Minimum%20Cost%20Flow%20Part%20One:%20Key%20Concepts

  https://bit.ly/32tElum

Demonstrações

- **Minimum-Cost Flows**

  https://jeffe.cs.illinois.edu/teaching/algorithms/notes/G-mincostflow.pdf

  https://bit.ly/35a51kL

Referências

Problemas para treino

- **Hackerearth:**

  https://www.hackerearth.com/practice/algorithms/graphs/minimum-cost-maximum-flow/practice-problems/

  https://bit.ly/3tYGgmc

- **Codechef:**

  https://www.codechef.com/tags/problems/min-cost-flow

  https://bit.ly/3H2JrNb

Referências

Tópicos Avançados

- **Potenciais:** [Tutorial] Graph Potentials, Johnson's Algorithm, and Min Cost Max Flow

  https://codeforces.com/topic/96440/en2

  https://bit.ly/3KK15HG