Interlúdio entre D&C e algs. gulosos p/ tratar de grafos e algs. aleatorizados

Grafos e o Problema do Corte Mínimo

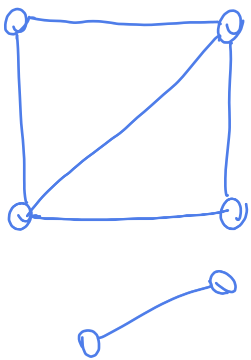
Grafos são uma estrutura matemática muito estudada

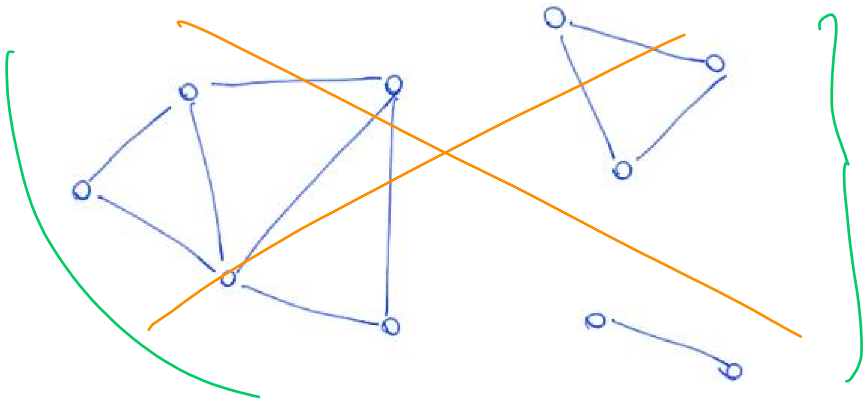
- e um Tipo Abstrato de Dado (TAD) usado para
  - representar relações entre elementos de um conjunto.
- Como todo TAD, precisa ser implementado por alguma estrutura de dados.

→ representam redes reais, conceituais ou implícitas

Grafos são formados por dois componentes: → $G = (V, E)$

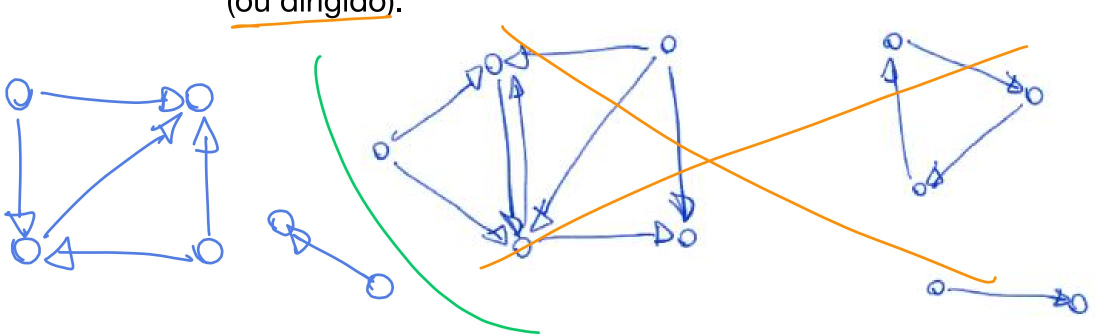
- Um conjunto de vértices (ou nós) V, e um conjunto de pares de vértices E.
- Se estes pares são não ordenados
  - os chamamos de arestas e o grafo é dito não orientado.





desenhar

- Se os pares são ordenados os chamamos arcos e o grafo é dito orientado (ou dirigido).



desenhar

Em geral, grafos são representados compactamente como G = (V, E), e usamos

- n = |V| para indicar o número de vértices,
- m = |E| para indicar o número de arestas.

$m = |E|$
$n = |V|$

Grafos são relevantes tanto na matemática quanto na computação, pois

- conseguem modelar uma grande variedade de cenários, como:
  - Redes físicas (elétrica, comunicações, transportes),
  - redes conceituais (Web, sociais, lógicas, biológicas),
  - estruturas como listas encadeadas e árvores,
  - relações de dependência ou interação (grafo de filmes e atores),
  - mapas, etc.
- Quiz1: quem são os vértices e as arestas (ou arcos) em cada cenário?
  - Quais cenários são não-orientados e quais são orientados?

De modo mais geral, grafos modelam relações entre pares de um mesmo conjunto,

- ou relações entre pares de conjuntos relacionados,
    - o que abre uma imensa gama de possibilidades.

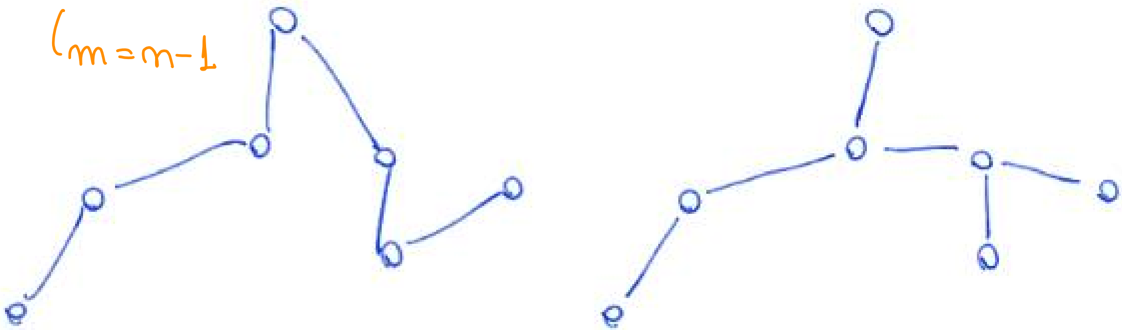
Grafos podem ser densos ou esparsos,

- o que diz respeito ao número de arestas que estes possuem.

Um grafo não orientado, conexo e sem arestas múltiplas possui:

- No mínimo n - 1 arestas, caso em que o grafo é uma árvore

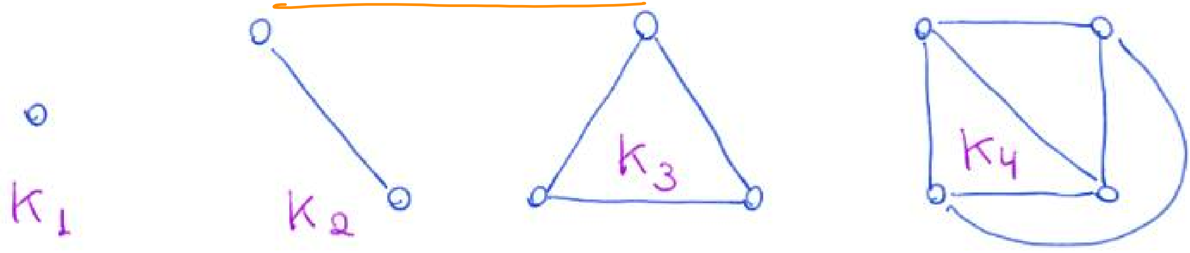
($m = n - 1$



desenhar

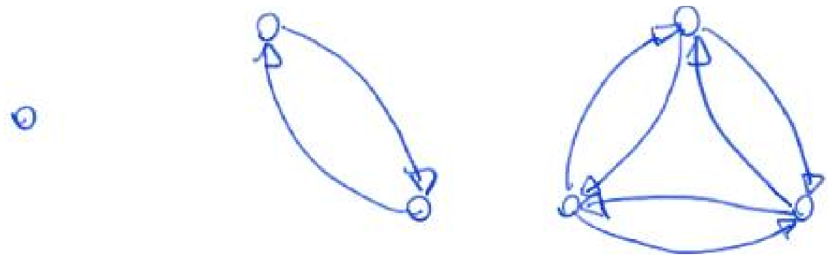
    - Quiz2: Por que? Dica: comece sem arestas e vá adicionando até ter um grafo conexo.

- No máximo (n escolhe 2) = $n(n-1)/2$ arestas, caso de um grafo completo.



desenhar

  - Um grafo orientado completo tem $n(n-1)$ arcos,
    - i.e., o dobro do não orientado, pois entre cada par de vértices podem ter dois arcos em sentidos opostos.

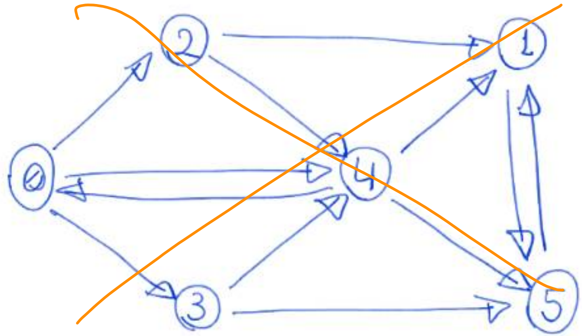


Assim, o número de arestas de um grafo varia entre $O(n)$ até $O(n^2)$.

- Dizemos que um grafo é esparso quando seu número de aresta
  - está próximo a n ou até n log n.
- Dizemos que ele é denso quando o número de arestas
  - está próximo de n^2 ou pelo menos superior $n^{3/2} = n * n^{1/2}$.
- Embora, onde passa a linha exatamente seja arbitrário.

Existem duas implementações principais para grafos,

- i.e., duas estruturas de dados usadas para representá-los.
- Em ambas, os vértices são rotulados por inteiros não negativos.

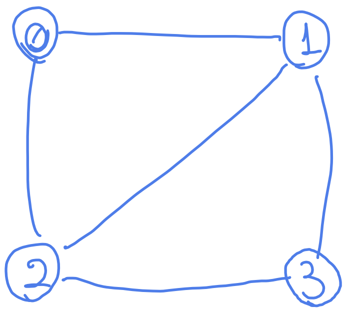


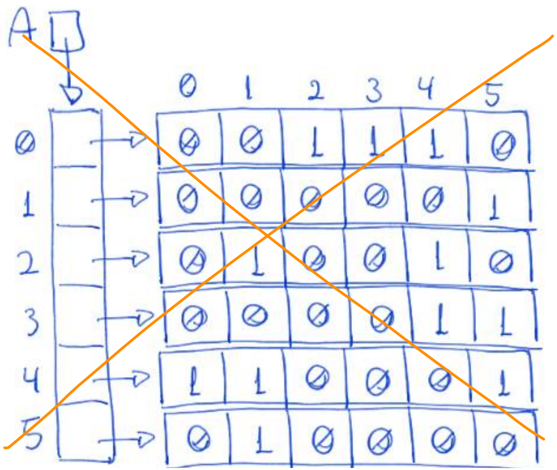
Matriz de adjacência é uma implementação que utiliza uma matriz A

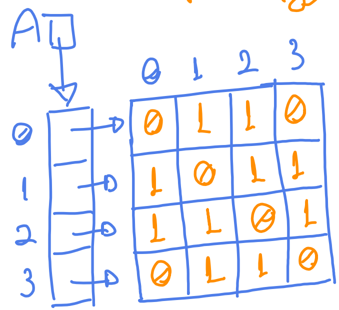
- de 0s e 1s com n linhas e n colunas sendo que o valor da célula A[i][j]
    - indica se existe uma aresta/arco entre os vértices i e j.

memória $\Theta(n^2)$

acesso a um par (i,j) $\Theta(1)$







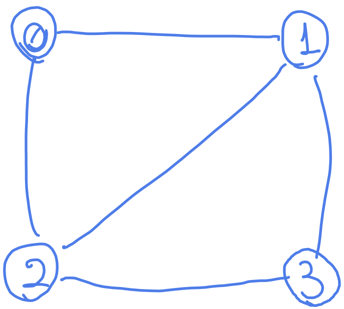
acesso a todos os vizinhos de um nó $\Theta(n)$

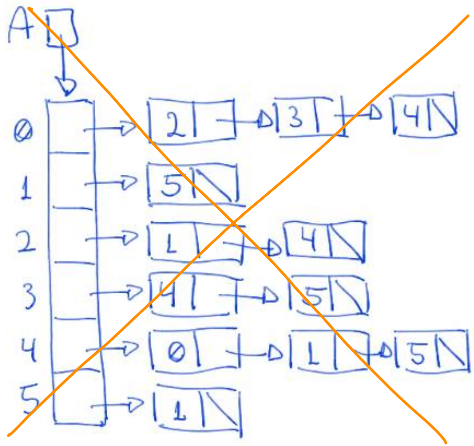
- Assim, a linha i da matriz A representa o leque de saída do vértice i
    - e a coluna j de A representa o leque de entrada do vértice j.
- A diagonal da matriz é preenchida por 0s, pois não temos auto-laços.
- Se o grafo não for orientado, a matriz é simétrica, i.e., A[i][j] = A[j][i].

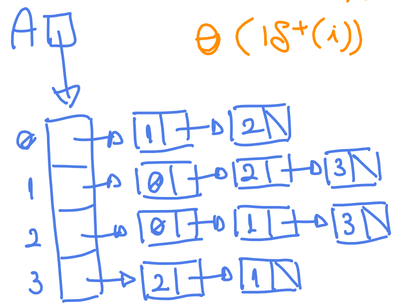
Listas de adjacência é uma implementação que utiliza
- um vetor A de apontadores de vértices de tamanho n
    - e para cada vértice i temos uma lista ligada iniciada em A[i],
        - com os destinos das arestas que têm origem em i.

☐ memória $\Theta(n+m)$

☐ acesso arbitrário a um par (i,j) $\Theta(|\delta^+(i)|)$







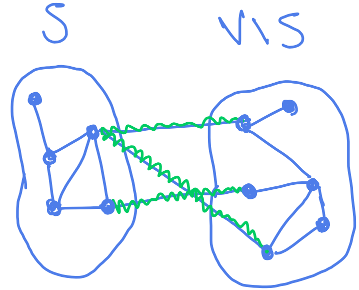
☐ acesso a todos os vizinhos de um nó i $\Theta(|\delta^+(i)|)$

- Se o grafo não for orientado, dada uma aresta {i, j},
    - temos que j será inserido na lista de i
        - e i será inserido na lista de j.

Corte em Grafos

- definição e exemplos

Def.: um corte S de um grafo $G = (V, E)$ é um subconjunto $S \subset V$ e $S \neq \emptyset$, i.e., S separa os vértices de V em duas partes

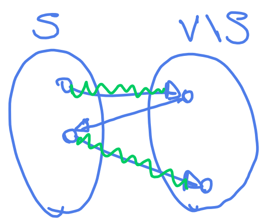


A fronteira do corte S ($\delta(S)$) são as arestas de E que tem um extremo em S e outro em $V \setminus S$

Quiz: qual o # de cortes de um grafo c/ n vértices? R: $2^n - 2$

todos os conjuntos exceto V e $\emptyset$

Em grafos dirigidos a fronteira costuma ser as arestas c/ origem em S e destino em $V \setminus S$



Quiz: todo conjunto de arestas é fronteira de algum corte? Ou ao menos está na fronteira de algum corte?

Problema do Corte Mínimo

- com exemplos

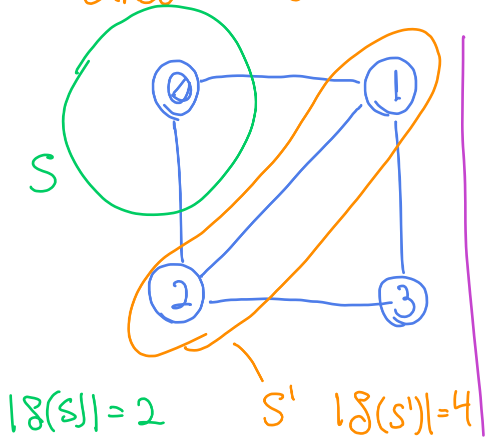
⊛ Aplicações: - encontrar gargalos/fraquezas p/ desconectar uma rede
- encontrar clusters razoavelmente isolados em redes sociais → testar produtos
- segmentar objetos em imagens

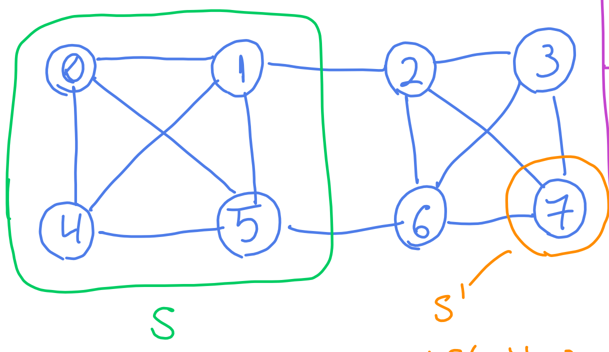
Entrada: um grafo não-orientado $G = (V, E)$

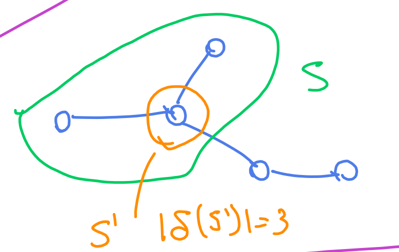
Solução: um corte $S \subset V$: $|\delta(S)|$ é mínima, i.e., o # de arestas atravessando a fronteira de S é $\leq |\delta(S')|$ p/ qualquer S' c/ $\emptyset \neq S' \subset V$

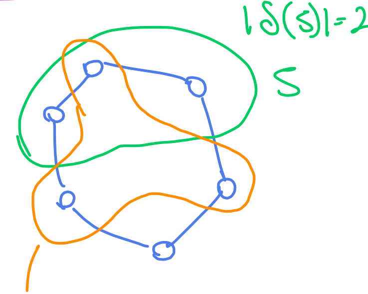
$|\delta(S)|$ = # de arestas atravessando a fronteira de S

Encontrar o corte mínimo nos seguintes grafos:









⊛ Obs: o corte mínimo sempre induz um subgrafo conexo. Quiz: Por quê?

Algoritmo de Contração Probabilística de Karger

- pseudocódigo e exemplos

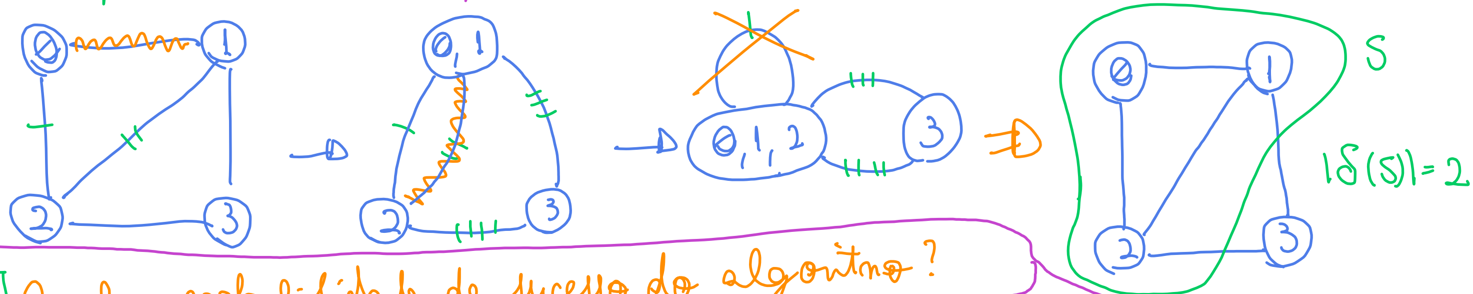
⊛ Curiosidade: no exemplo a seguir o corte $S = \{0,3\}$ não pode ocorrer

⊛ Intuição: tem mais chances de escolher arestas de cortes grandes, pois estes tem mais arestas

enquanto # de vértices > 2:

- escolha uma aresta $e = \{u,v\}$ c/ prob. uniforme
- contraia os vértices u e v, mantendo arestas múltiplas, mas removendo auto-laços que surjam

devolva o corte S correspondente a um dos vértices restantes



⊛ Qual a probabilidade de sucesso do algoritmo?