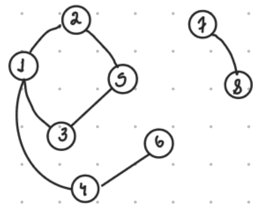
Conectividade

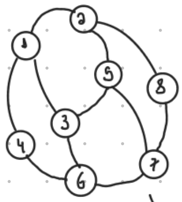
Um grafo não-direcionado é conexo, se todos os vértices forem alcançáveis a partir de qualquer outro.

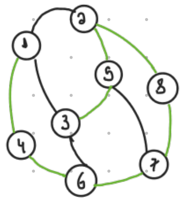
Um grafo não-direcionado é desconexo quando não existir um caminho entre algum par de vértices. ⟷ o fecho transitivo de algum vértice for diferente do conjunto de vértices.



Num grafo não direcionado conexo é possível fazer um passeio fechado que inclua todos os outros vértices.

Exemplo: 3, 5, 2, 8, 7, 6, 4, 1, 3





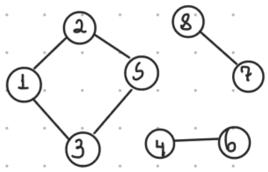
Fecho transitivo direto de qualquer vértice é o conjunto de vértices.

Componente Conexo

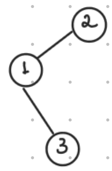
Num grafo não-direcionado, seus componentes conexos são os subgrafos maximais que são conexos.

Um subgrafo maximal conexo é um dos maiores subgrafos (em número de vértices e arestas) que são conexos.

G

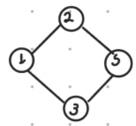


H_1



subgrafo conexo mas não maximal

H_2



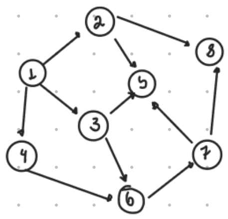
subgrafo conexo maximal

Grafo Subjacente

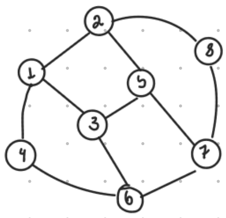
associado

grafo não-direcionado resultante de um grafo direcionado em que a direção das arestas é removida.

deve ser simples.



G direcionado

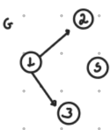


G subjacente

G. direcionado

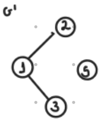
Um grafo direcionado G é conexo se o seu grafo associado (subjacente) for conexo. ⟶ A mesma lógica se aplica para G ser desconexo

G



associado

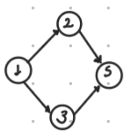
G'



desconexo

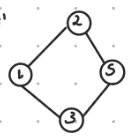
⟶ G é desconexo

G



associado

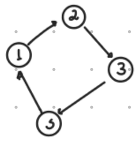
G'



G' conexo

⟶ G é conexo

Simplesmente Conexo → O associado é conexo



Semi fortemente conexo → Para todo par de vértice pelo menos um deles é alcançado a partir de outro

(u,v) existente $\exists$ path (u,v) ou (v,u)

Fortemente Conexo → Todos os vértices são mutualmente alcançáveis.

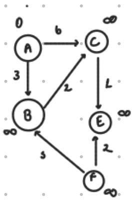
$\forall\ u,v \in V \mid \exists\ (u,v)$ e (v,u)

DIJKSTRA

A cada iteração, calcula-se uma estimativa da distância entre os vértices já explorados e seus vizinhos não explorados. (analisar apenas as arestas de corte).

1) $\forall\ v \in V$, $Dist[v] = \infty$ e $pred[v] = -1$
2) Selecionar um vértice $v \in V$ = início
3) $dist[início] \leftarrow 0$
4) Enquanto existirem vértices para ser processado
   5) v = arg Min (dist $[v]$ | v ainda não processado) → A
   6) Marcar v como processado
   7) Para todos os vizinhos w de v
      8) Se $dist[w] > dist[v] + peso(v,w)$
         $dist[w] = dist[v] + peso(v,w)$
         $pred = [v]$

Execução



1)

	A	B	C	D	E	F
dist	∞	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

2) $v \leftarrow A$

3) $dist[A] = 0$

4) vértices não processados [A, B, C, D, E, F]
   5) $v \leftarrow A$
   6) vértices não processados [~~A~~, B, C, D, E, F]
   7) vizinhos de v = C, B
   8) $dist[B] = \infty > (0 + 3)$? Sim
   9) $dist[B] = 3$
   10) $pred[B] = A$

Após primeira execução

	A	B	C	D	E	F
dist	0	3	∞	∞	∞	∞
pred	-1	A	-1	-1	-1	-1