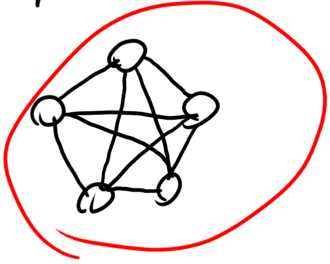
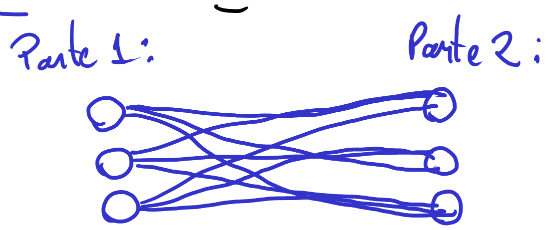
## 1.2.18 Planaridade

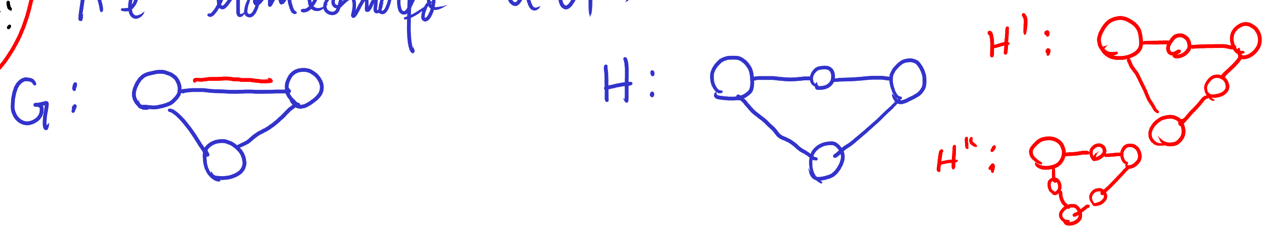
$K_5$: um grafo completo com 5 vértices



$K_{3,3}$: um grafo bipartido com 3 vértices em cada parte, completo




**Homeomorfo:** H é homeomorfo a G:

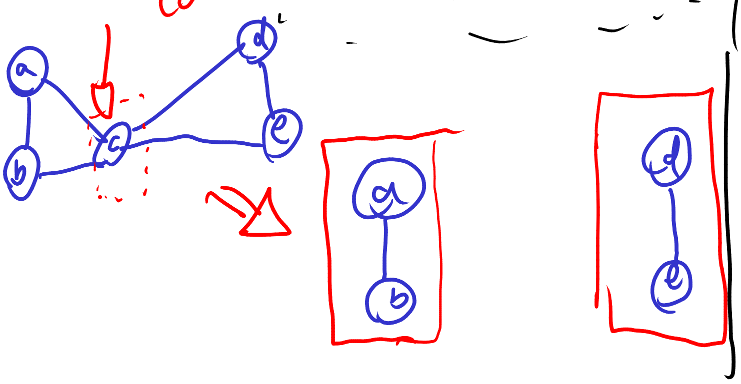



Harary: (Pag 17 e 18)

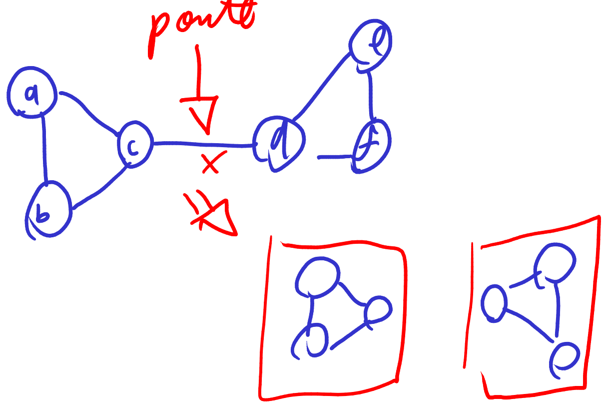
| 1.2.19 Vértice de corte | 1.2.20 Ponte |
|---|---|
| Vértice | Aresta |

Se sua remoção causaria o incremento no número de componentes desconexas

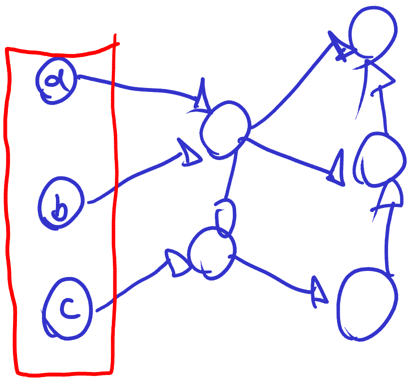
vértice de corte




ponte

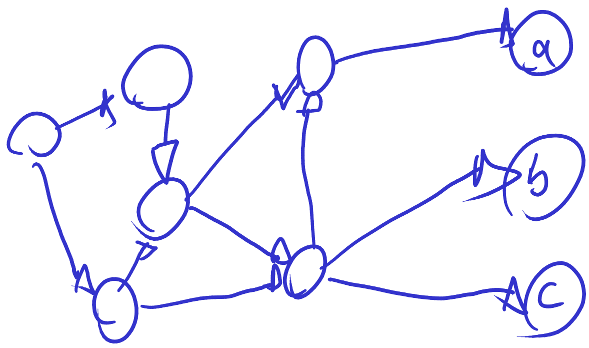


## 1.2.21 Base




$B = \{a, b, c\}$

## 1.2.22 Anti-Base




$A = \{a, b, c\}$

# Aquecimento p/ Q2

1) • conjunto $L = \{(a, b, i)\}$ a e b são os sites visitados, i é identidade do usuário.

$\boxed{\text{a antes de b}}$

## Algoritmo Q2.1

Entrada: o conjunto L, identificador "k" do usuário

$T'$    $n = |L|$

1. $V \leftarrow \{\}$ — 1
2. $A \leftarrow \{\}$ — 1
3. foreach $(a, b, i) \in L$ do
4.     if $i = k$ then — 1
5.         $V \leftarrow V \cup \{a, b\}$ — 1
6.         $A \leftarrow A \cup \{(a, b)\}$ — 1
7. $G \leftarrow (V, A)$ — 1    $|A| = n$ ; $|V| = 2n$    $\Theta(|V| + |A|)$
8. $S \leftarrow$ Algoritmo 15(G) // Pag 68 → $T''(n) \le 2n + n$

$T''(n)$

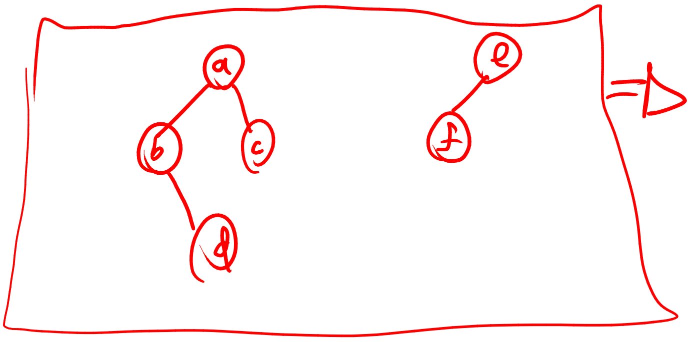
9. ...

$T'(n) = 2 + 1 + 3n$

$T'(n) \in O(n)$

$T''(n) \in O(n)$

$\boxed{T'(n) + T''(n) \le c_1 \cdot n + c_2 \cdot n} \in O(n)$



$\Rightarrow \{\{a,b,c,d\},\{e,f\}\}$

$S_a = N$

$S_b = a$

$S_c = a$

$S_d = b$

$S_e = N$

$S_f = e$

$R_a = \{a\} b, c, d$

$R_b = \{b\} a, c, d$

$R_c = \{c\} a, b, d$

$R_d = \{d\} b, a, c$

$R_e = \{e\} f$

$R_f = \{f\} e$

$\{\{a,b,c,d\},\{e,f\}\}$

---

9. $R_x = \{x\} \quad \forall x \in V \rightarrow |V| = 2n \quad |A| = n$

$T'''(n)$ — $2n \times$

10. foreach $u \in V$ do — $2n(4+2n)$
11. if $S_u \neq n$ then — 1
12. $v := S_u$ — 1
13. $U := R_u \cup R_v$ — 1
14. foreach $z \in U$ do — $2n$
15. $\quad R_z \leftarrow U$ — 1

$T'''(n) \leq 8n + 4n^2$

$T'''(n) \in O(n^2)$

$T^{IV}(n)$

16. $X \leftarrow \{\}$ — 1
17. foreach $v \in V$ do — $2n$ ($|V| = 2n$)
18. $\quad X \leftarrow X \cup \{R_v\}$ — 1
19. return $X$.

$T^{IV}(n) \leq 2n$

$T^{IV}(n) \in O(n)$

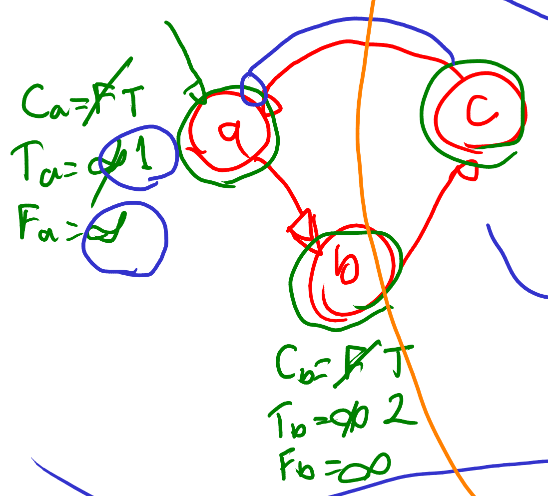
$T(n) = O(n) + O(n) + O(n^2) + O(n) \in O(n^2)$

Aq Q2.2

- $T = \{t_1, t_2, \ldots, t_n\}$ conj. de tarefas
- $R = \{(t_i, t_j)\}$ $t_i$ depende de $t_j$

invertida em relação ao alg. de O.T. (Pag 74)

$T = 1$




B Quando $T_a \neq \infty$ e $F_a = \infty$

"a"

Para preparar o grafo, cada vértice seria uma tarefa, e cada arco (x, y) seria composto por uma dependência entre tarefas: y depende de x.

Adaptar o algoritmo 19 (Pag. 74):
- Na linha 5, adicionar a seguinte instrução:

if $T_u \neq \infty$ and $F_u = \infty$ then
  return False

- Depois da linha 9, adicionar:

return true

- Substituir o que existe dentro do desvio condicional da linha 5 por:

$r \leftarrow$ DFS-Visit-OT(G, u, C, T, F, tempo, O)
if r = false then
  return False

Adaptar o algoritmo 18 (Pag. 74):
- Substituir o que existe dentro do desvio condicional da linha 7 por:

$r \leftarrow$ DFS-Visit-OT(G, u, C, T, F, tempo, O)
if r = False then
  return (false, null)

- Substituir a instrução da linha 9 por:

(Foreach o ∈ O do
  print(o + ",")) – |V|
return (true, O)