

Algoritmos

Pedro Hokama

Fontes

- [clrs] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest, Ronald L. Rivest e Clifford Stein.

- [timr] Algorithms Illuminated Series, Tim Roughgarden

Apresentação Baseada:

- Stanford Algorithms

<https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V>

<https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZ17A5HEK6>

- Conjunto de Slides dos Professores Cid C. de Souza, Cândida N. da Silva, Orlando Lee, Pedro J. de Rezende

- Conjunto de Slides do Professores Cid C. de Souza para a disciplina MO420

Qualquer erro é de minha responsabilidade.

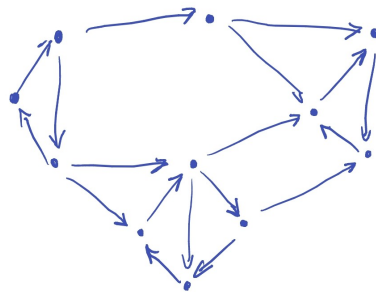
1 / 9

2 / 9

Componentes fortemente conexas - Strongly Connected Components

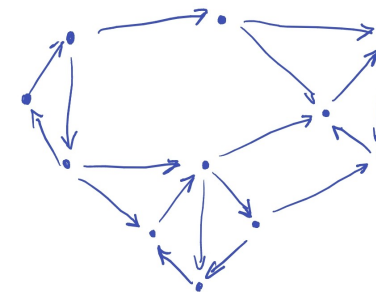
Definição

Componentes Fortemente Conexas de um Grafo Direcionado são classes de equivalência da relação $u \sim v \iff$ existe um $u - v$ caminho, e um $v - u$ caminho em G . (\sim é uma relação de equivalência)



3 / 9

DFS para encontrar as SCCs?



4 / 9

Algoritmo de Kosaraju

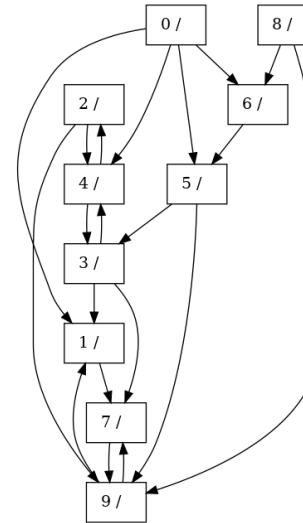
O algoritmo de Kosaraju faz duas passadas no DFS (com pequenas modificações), tem complexidade $O(|V| + |E|)$.

Ideia do algoritmo:

- Reverter todos os arcos de G e chamar de G^{rev}
- DFS-Loop em G^{rev} (salvando os tempos de termino)
- DFS-Loop em G (na ordem decrescente de tempos de termino da primeira passada)

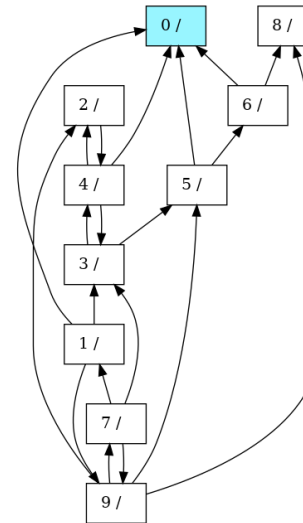
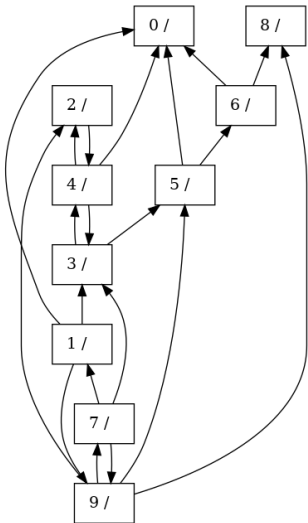
Observações:

- Computaremos líderes para cada vértice, vértices com o mesmo líder estarão na mesma SCC.



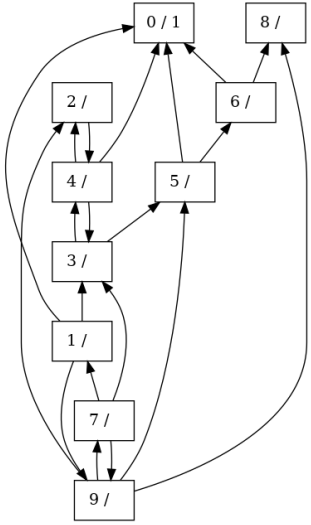
5 / 9

6 / 9

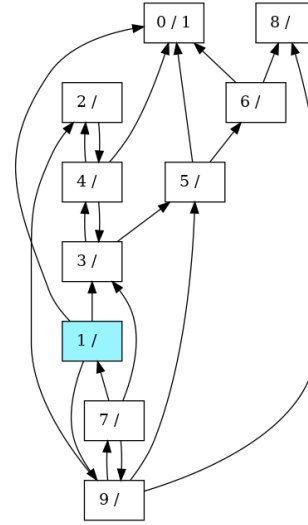


7 / 9

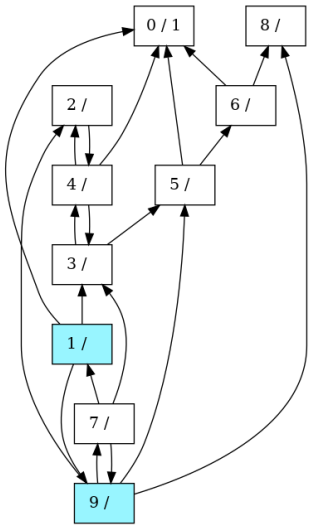
8 / 9



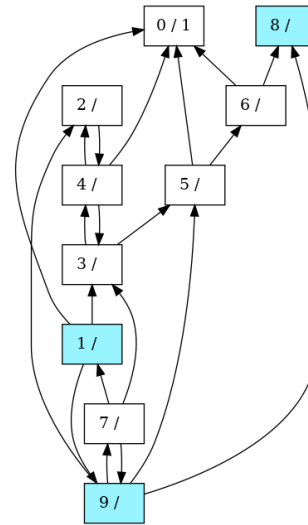
9 / 9



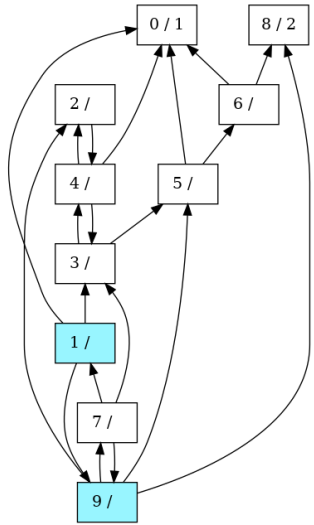
10 / 9



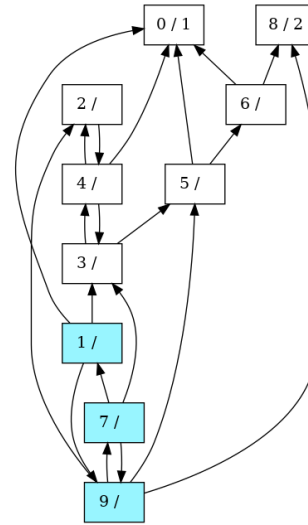
11 / 9



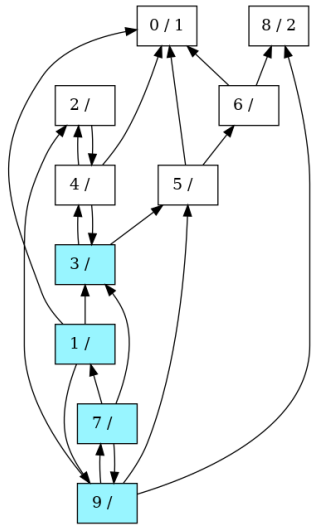
12 / 9



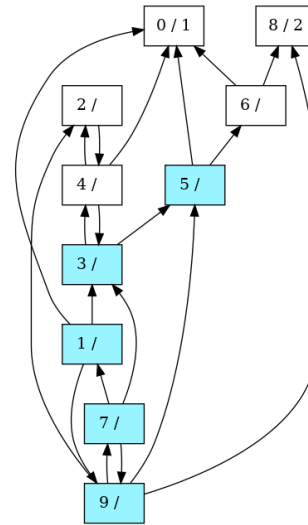
13 / 9



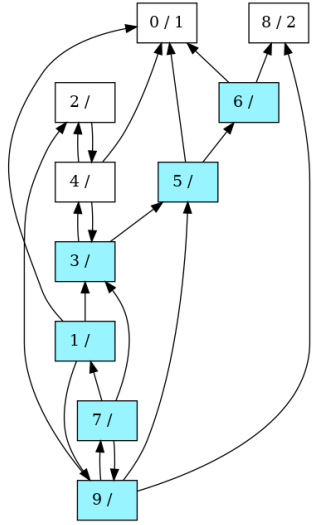
14 / 9



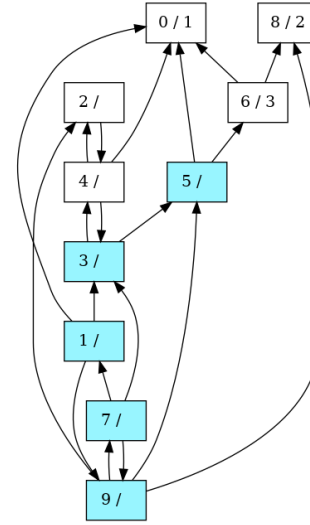
15 / 9



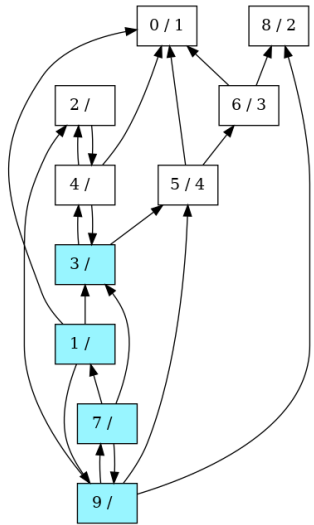
16 / 9



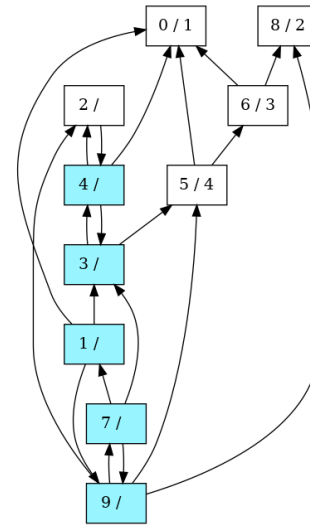
17 / 9



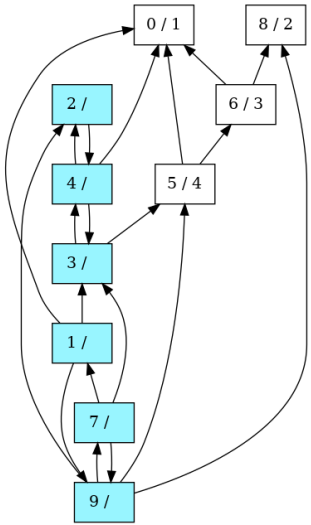
18 / 9



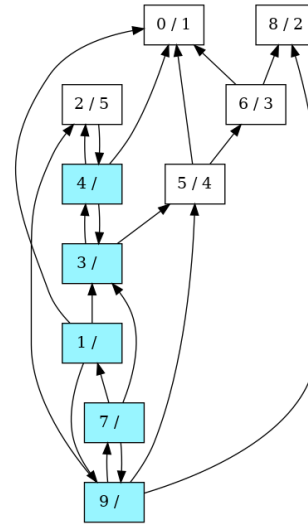
19 / 9



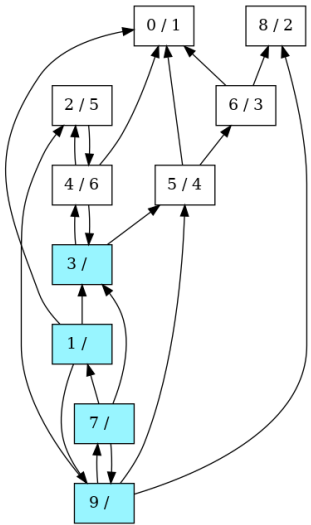
20 / 9



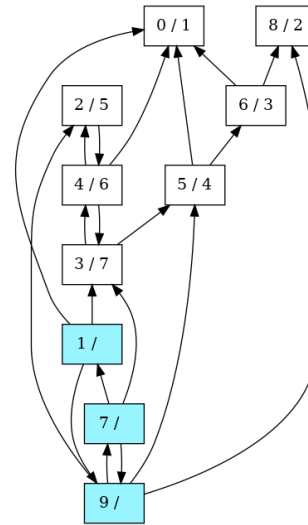
21 / 9



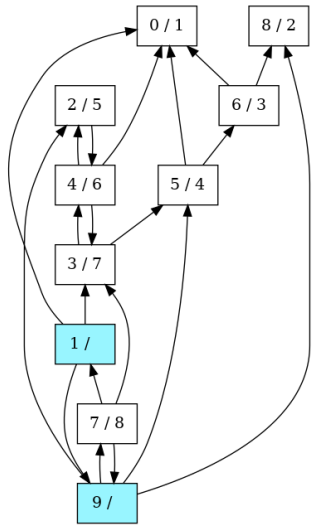
22 / 9



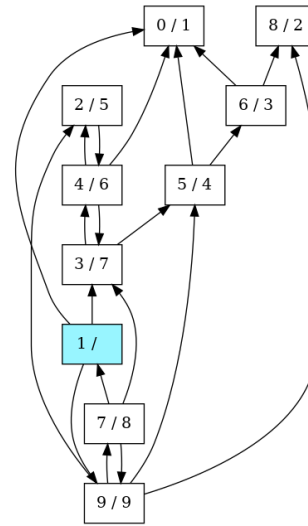
23 / 9



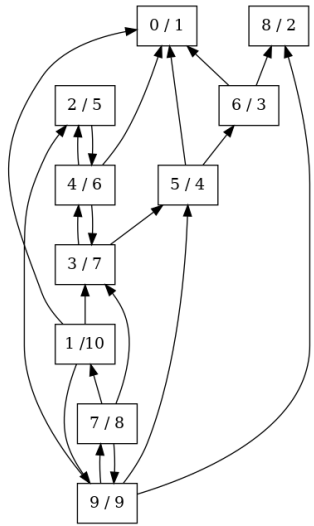
24 / 9



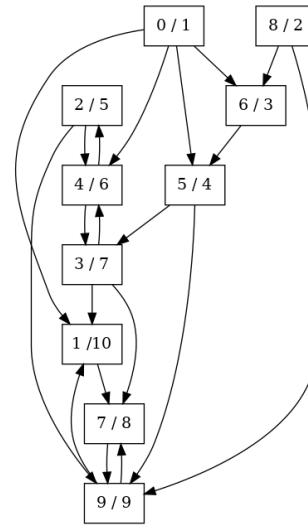
25 / 9



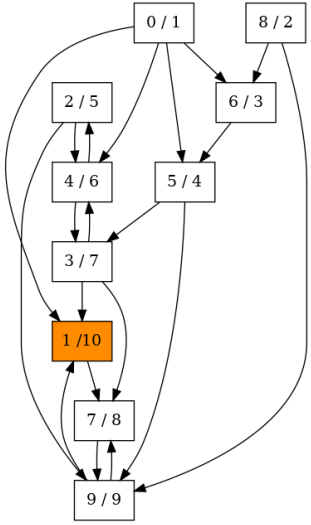
26 / 9



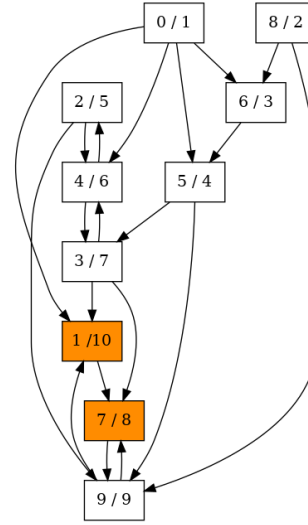
27 / 9



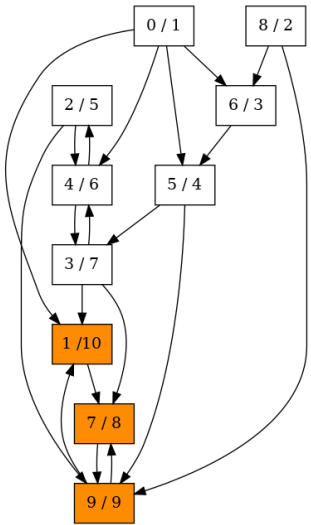
28 / 9



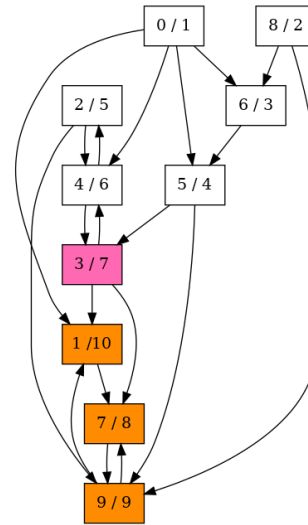
29 / 9



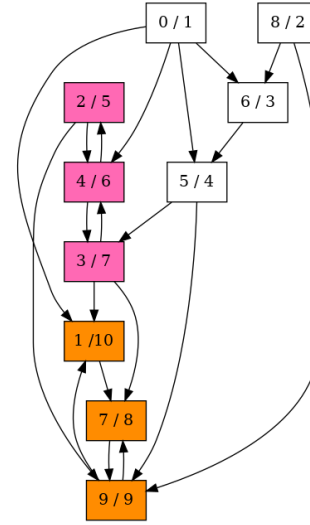
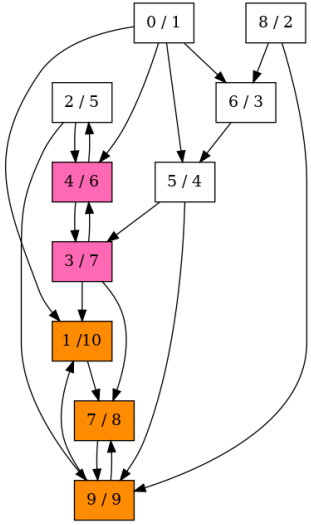
30 / 9



31 / 9

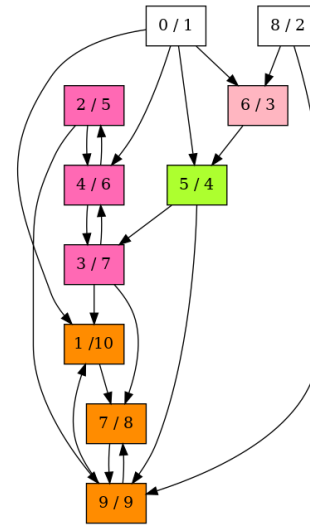
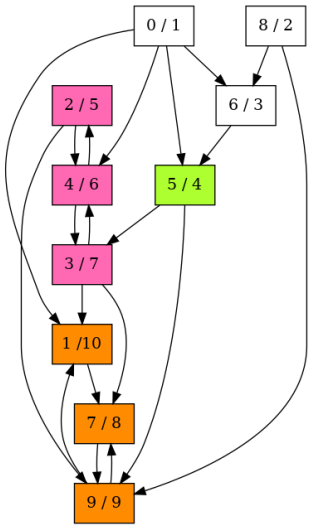


32 / 9



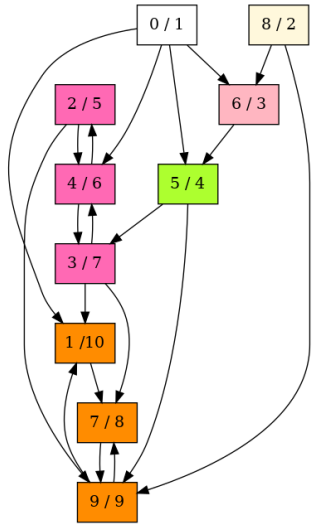
33 / 9

34 / 9

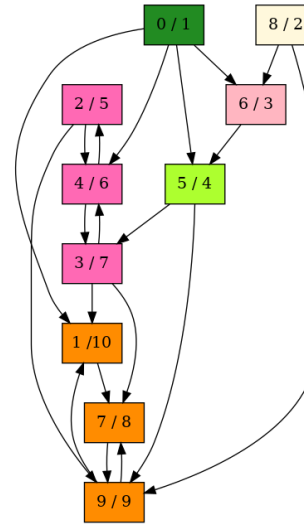


35 / 9

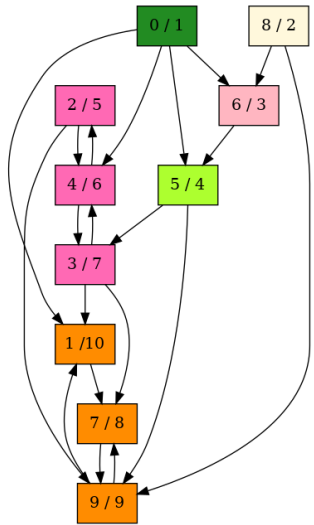
36 / 9



37 / 9



38 / 9



39 / 9

Algoritmo 1: DFS-Loop

Entrada: Um Grafo G

- 1 Marcar $v \in V$ como não-visitado;
 - 2 $T = 0$;
 - 3 $L = NULL$;
 - 4 Assuma que os vértices estão rotulados de 1 até n ;
 - 5 **para** $v = n$ até 1 **faça**
 - 6 **se** v está não-visitado **então**
 - 7 $L = v$;
 - 8 $DFS(G, v)$;
-

Algoritmo 2: DFS

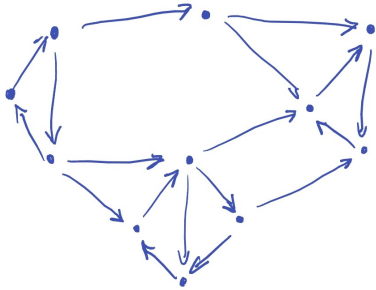
Entrada: Um Grafo G , e um vértice fonte s

- 1 Marcar s como visitado ;
 - 2 $lider(s) = L$;
 - 3 **para todo** arco (s, v) **faça**
 - 4 **se** v é não visitado **então**
 - 5 $DFS(G, v)$;
 - 6 $T ++$;
 - 7 $f(s) = T$;
-

40 / 9

Corretude do Kosaraju

- As SCCs de um grafo direcionado G correspondem a um "Meta-Grafo" acíclico em que:
- Os meta-vértices são as SCCs C_1, \dots, C_k de G .
- Existe um arco de $(C, C') \iff$ Existe $(i, j) \in A$ com $i \in C$ e $j \in C'$.



- O grafo é acíclico, senão cada C não seria uma SCC.

41 / 9

Lema 1

Considere duas SCC "adjacentes" em G , sendo que existe uma aresta de C_1 para C_2 . Seja $f(v)$ = tempo de termino do DFS-Loop no G^{rev} . Então:

$$\max_{v \in C_1} f(v) < \max_{v \in C_2} f(v)$$

Prova: Seja v o primeiro vértice visitado de $C_1 \cup C_2$, na primeira passagem do algoritmo.

Note que em G^{rev} a aresta considerada está invertida.

- 1 Caso 1: $v \in C_1$, então todos os vértices de C_1 serão visitados antes de qualquer vértice de C_2
- 2 Caso 2: $v \in C_2$, eventualmente algum vértice de C_1 é alcançado e todo C_1 é explorado antes de C_2 terminar.

42 / 9

Corolário 1

O maior valor de f deve ficar em um SCC sorvedouro.

Corretude do Kosaraju

- Como o algoritmo começa a executar em uma SCC sorvedouro, ele encontra toda a componente sem sair dela.
- A partir daí os vértices dessa SCC não é mais considerado e a próxima SCC sorvedouro é executada.
- Dessa forma cada SCC é encontrada separadamente □.

43 / 9