

UNIFEI - UNIVERSIDADE FEDERAL DE ITAJUBÁ
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO



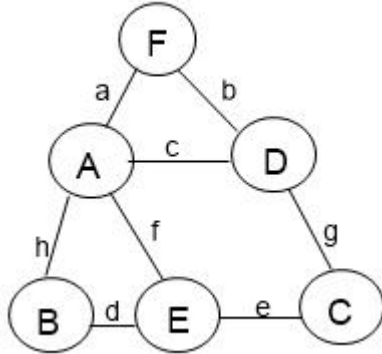
SIN110 - ALGORITMOS E GRAFOS
RESOLUÇÃO DOS EXERCÍCIOS E05 DO DIA 18/09/2015

Exercícios E05 – 18/09/15

Aluna: Karen Dantas

Número de matrícula: 31243

1) Grafo:



Busca em largura a partir do vértice F:

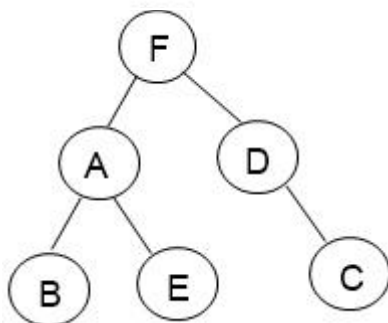
Vértice	Cor(u)	Predecessor(u)	Dist(u)
A	⊥ / -e / -p	F	1
B	⊥ / -e / -p	A	2
C	⊥ / -e / -p	D	2
D	⊥ / -e / -p	F	1
E	⊥ / -e / -p	A	2
F	⊥ / -e / -p	-	0

Filas:

u = F F- A – D
u = A A- D – E - B
u = D D – E – B – C
u = E E – B – C
u = B B – C
u = C C

Ordem de visita: F- A - D - E - B – C

Arborescência:



2) Algoritmo:

Encontra_MenorCaminho_Duende(G, x)

```
1.   para u ← 1 ate n faça
2.       cor[u] ← BRANCO
3.       dist[u] ← ∞
4.   cor[x] ← CINZA
5.   dist[x] ← 0
6.   menor ← ∞
7.   Q ← Inicializa-Fila (Q,x)
8.   enquanto Q ≠ ∅ faça
9.       u ← Primeiro-da-Fila (Q)
10.      para cada v em Adj[u] faça
11.          se cor[v] = BRANCO e valor[v] ≠ 2
12.              entao cor[v] ← CINZA
13.              dist[v] ← dist[u]+1
14.              se dist[v] < menor e valor[v]=0
15.                  então menor← dist[v]
16.              Insira-na-Fila (Q,v)
17.      Remova-da-Fila (Q)
18.      cor[u] ← PRETO
19.   devolve menor
```

Correção: As condições de parada das linhas 1, 8 e 10 são válidas, logo, a execução do algoritmo tem fim.

O algoritmo funciona, pois, o *loop* da linha 1 faz uma inicialização das variáveis cor e dist de todos os vértices. O vértice de partida (x) é inicializado com a cor cinza, sua dist com 0 e a variável menor é inicializada com infinito. É inserido na fila Q o vértice x e, na linha 8, entra-se em um loop o qual só terminará quando a fila estiver vazia. Logo após, a variável u é inicializada com o primeiro elemento da fila e na linha 10 entra-se em outro loop que irá percorrer os vértices adjacentes do vértice u. Dentro desse loop é verificado se cor de v é branco e se o valor de v é diferente de dois. Se cor de v for diferente de branco (quer dizer que já foi visitado) e/ou o valor de v for igual a dois (indica que o salão possui parede de cristal), nenhuma operação é executada e o *loop* continua. Caso contrário, a cor de v fica cinza, o dist de v recebe o dist de u incrementado de um, e, se o dist de v for menor que o número contido na variável menor e o valor de v for igual a zero, significa que foi encontrada uma saída e que a distância foi a menor encontrada até então. Na linha 16, é inserido o vértice v no fim da fila Q e o *loop* continua. Quando o *loop* termina é removido o primeiro elemento da fila, a cor de u fica preto e o *loop* da linha 8 continua enquanto houver elementos na fila. Por fim, é retornado o número mínimo de salões que o duende deverá percorrer para sair da caverna são e salvo.

Tempo de execução:

Encontra_MenorCaminho_Duende(G, x)	Contagem
1. para u ← 1 ate n faça	n+1
2. cor[u] ← BRANCO	n
3. dist[u] ← ∞	n
4. cor[x] ← CINZA	1
5. dist[x] ← 0	1
6. menor ← ∞	1
7. Q ← Inicializa-Fila (Q,x)	1
8. enquanto Q ≠ ∅ faça	n+1
9. u ← Primeiro-da-Fila (Q)	n
10. para cada v em Adj[u] faça	v+1
11. se cor[v] = BRANCO e valor[v] ≠ 2	v
12. entao cor[v] ← CINZA	v
13. dist[v] ← dist[u]+1	v
14. se dist[v] < menor e valor[v]=0	v
15. então menor← dist[v]	v
16. Insira-na-Fila (Q,v)	v
17. Remova-da-Fila (Q)	n
18. cor[u] ← PRETO	n
19. devolve menor	1

Para o cálculo foi considerado número de vértices (n) e número de adjacentes (v).

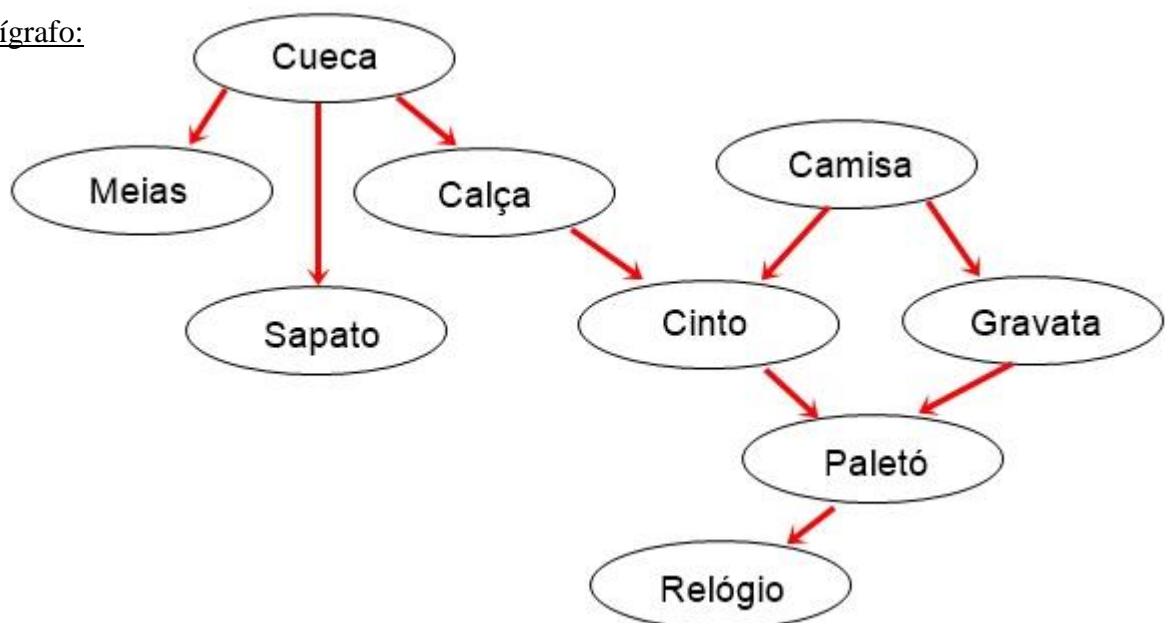
$$F(t) = n+1+n+n+1+1+1+1+n+1+n+v+1+v+v+v+v+v+v+n+n+1$$

$$F(t) = 7n + 7v + 8$$

Complexidade assintótica: $O(n+v)$.

Logo, o consumo de tempo do algoritmo é de $O(n+v)$.

3) Dígrafo:

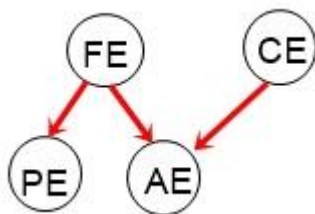


A partir da representação acima do problema, irei simular a execução do algoritmo de ordenação topológica para descobrir a ordem correta que Mr. Bean deverá seguir.

Vértice	Cor(u)	K	Topo(K)
Cueca	b / p	9	Sapato
Meias	b / p	8	Meias
Calça	b / p	7	Relógio
Sapato	b / p	6	Paletó
Camisa	b / p	5	Cinto
Cinto	b / p	4	Calça
Gravata	b / p	3	Cueca
Paletó	b / p	2	Gravata
Relógio	b / p	1	Camisa

A partir do vetor gerado, sabe-se que a sequência que o Mr. Bean deverá seguir é:
 Camisa → Gravata → Cueca → Calça → Cinto → Paletó → Relógio → Meias → Sapato

4) Dígrafo:



- i) Para descobrir a ordem que as tarefas devem ser executadas irei simular a execução do algoritmo de ordenação topológica.

Vértice	Cor(u)	K	Topo(K)
FE	b / p	4	PE
PE	b / p	3	AE
AE	b / p	2	FE
CE	b / p	1	CE

A partir do vetor gerado, sabe-se que a ordem que deverá ser seguida é:
 CE → FE → AE → PE

ii) O menor tempo possível para construir a cerca é representado pela sequência $CE \rightarrow FE \rightarrow AE \rightarrow PE$, que, respectivamente, gastam 2, 3, 2, 4 unidades de tempo gerando um total de 11 unidades de tempo.

5) Função que calcula o diâmetro de qualquer grafo conexo:

Diametro_grafo_conexo (G,x)

```
1.  para u ← 1 ate n faça
2.      cor[u] ← BRANCO
3.      dist[u] ← ∞
4.  cor[x] ← CINZA
5.  dist[x] ← maior ← 0
6.  Q ← Inicializa-Fila (Q,x)
7.  enquanto Q ≠ ∅ faça
8.      u ← Primeiro-da-Fila (Q)
9.      para cada v em Adj[u] faça
10.         se cor[v] = BRANCO
11.             entao cor[v] ← CINZA
12.             dist[v] ← dist[u]+1
13.             se dist[v] > maior
14.                 então maior ← dist[v]
15.             Insira-na-Fila (Q,v)
16.      Remova-da-Fila (Q)
17.      cor[u] ← PRETO
18.  devolve maior
```

Foi criada uma variável maior que receberá a maior distância existente entre o vértice x e os demais vértices do grafo. E essa maior distância, que é o diâmetro do grafo, é retornada pela função ao fim de sua execução.