

Trabalho de Fluxo em Redes e Árvores

O problema de fluxo em redes envolve determinar a quantidade máxima de fluxo que pode ser enviado através de uma rede. Já as árvores em grafos são subgrafos sem ciclos. Aplicações do fluxo em redes incluem roteamento de tráfego em redes de computadores e fluxo de produtos em cadeias de suprimentos. Algoritmos como Ford-Fulkerson e Edmonds-Karp são usados para resolver problemas de fluxo em redes. As árvores em grafos têm aplicações em estruturas de dados, hierarquias de organizações e algoritmos de otimização. Algoritmos como busca em largura, busca em profundidade, Prim e Kruskal são comumente usados para trabalhar com árvores em grafos.

Exercício no 12:

A tabela a seguir mostra a distância entre as doze cidades brasileiras que serão sedes na Copa Mundial de Futebol de 2014, sendo elas: Belo Horizonte (BHO), Brasília (BRA), Cuiabá (CBA), Curitiba (CTB), Fortaleza (FOR), Manaus (MAN), Natal (NAT), Porto Alegre (POA), Recife (REC), Rio de Janeiro (RJO), Salvador (SAL) e São Paulo (SPO). Determine o grafo de conexão mínima entre essas cidades utilizando o algoritmo de Kruskal e o de Prim.

BHO	BHO											
BRA	716	BRA										
CBA	1594	1133	CBA									
CTB	1004	1366	1679	CTB								
FOR	2528	2200	3406	3541	FOR							
MAN	3951	3490	2357	4036	5763	MAN						
NAT	2348	2422	3543	3365	537	5985	NAT					
POA	1712	2027	2206	711	4242	4563	4066	POA				
REC	2061	2135	3255	3231	800	5698	298	3779	REC			
RJO	439	1148	2017	852	2826	4374	2625	1553	2338	RJO		
SAL	1372	1446	2566	2385	1389	5009	1131	3090	897	1678	SAL	
SPO	586	1015	1614	408	3127	3971	2947	1109	2660	429	1962	

Resposta:

Para determinar o grafo de conexão mínima entre as cidades utilizando os algoritmos de Kruskal e Prim, primeiro precisamos transformar a tabela de distâncias em uma representação mais adequada. Podemos usar uma matriz de adjacência para isso, onde cada célula (i, j) representa a distância entre a cidade i e a cidade j.

	BHO	BRA	CBA	CTB	FOR	MAN	NAT	POA	REC	RJO	SAL	SPO
BHO	0	716	1594	1004	2528	3951	2348	1712	2061	439	1372	586
BRA	716	0	1133	1366	2200	3490	2422	2027	2135	1148	1446	1015
CBA	1594	1133	0	1679	3406	2357	3543	2206	3255	2017	2566	1614
CTB	1004	1366	1679	0	3541	4036	3365	711	3231	852	2385	408
FOR	2528	2200	3406	3541	0	5763	537	4242	800	2826	1389	3127
MAN	3951	3490	2357	4036	5763	0	5985	4563	5698	4374	5009	3971
NAT	2348	2422	3543	3365	537	5985	0	4066	298	2625	1131	2947
POA	1712	2027	2206	711	4242	4563	4066	0	3779	1553	3090	1109
REC	2061	2135	3255	3231	800	5698	298	3779	0	2338	897	2660
RJO	439	1148	2017	852	2826	4374	2625	1553	2338	0	1678	429
SAL	1372	1446	2566	2385	1389	5009	1131	3090	897	1678	0	1962
SPO	586	1015	1614	408	3127	3971	2947	1109	2660	429	1962	0

Agora podemos aplicar os algoritmos de Kruskal e Prim para obter o grafo de conexão mínima.

Algoritmo de Kruskal:

- Pegamos a lista de todas as conexões entre as cidades, ordenadas por distância.

NAT - REC (298)
CTB - SPO (408)
RJO - SPO (429)
BHO - RJO (439)
FOR - NAT (537)
BHO - SPO (586)
CTB - POA (711)
BHO - BRA (716)
FOR - REC (800)
CTB - RJO (852)
REC - SAL (897)
BHO - CTB (1004)
BRA - SPO (1015)
POA - SPO (1109)
NAT - SAL (1131)
BRA - CBA (1133)
BRA - RJO (1148)

BRA - CTB (1366)
BHO - SAL (1372)
FOR - SAL (1389)
BRA - SAL (1446)
POA - RJO (1553)
BHO - CBA (1594)
CBA - SPO (1614)
RJO - SAL (1678)
CBA - CTB (1679)
BHO - POA (1712)
SAL - SPO (1962)
CBA - RJO (2017)
BRA - POA (2027)
BHO - REC (2061)
BRA - REC (2135)
BRA - FOR (2200)
CBA - POA (2206)
REC - RJO (2338)
BHO - NAT (2348)
CBA - MAN (2357)
CTB - SAL (2385)
BRA - NAT (2422)
BHO - FOR (2528)
CBA - SAL (2566)
NAT - RJO (2625)
REC - SPO (2660)
FOR - RJO (2826)
NAT - SPO (2947)
POA - SAL (3090)
FOR - SPO (3127)
CTB - REC (3231)
CBA - REC (3255)
CTB - NAT (3365)
CBA - FOR (3406)
BRA - MAN (3490)
CTB - FOR (3541)
CBA - NAT (3543)
POA - REC (3779)
BHO - MAN (3951)
MAN - SPO (3971)
CTB - MAN (4036)
NAT - POA (4066)
FOR - POA (4242)
MAN - RJO (4374)

MAN - POA (4563)
MAN - SAL (5009)
MAN - REC (5698)
FOR - MAN (5763)
MAN - NAT (5985)

- Começamos com uma árvore vazia.
- Vamos adicionando as conexões mais curtas à árvore, uma por uma, desde que não formem um ciclo.
- Continuamos adicionando conexões até que todas as cidades estejam conectadas.

NAT - REC (298)
CTB - SPO (408)
RJO - SPO (429)
BHO - RJO (439)
FOR - NAT (537)
CTB - POA (711)
BHO - BRA (716)
REC - SAL (897)
BRA - CBA (1133)
BHO - SAL (1372)
CBA - MAN (2357)

O resultado final será uma árvore (grafo) que representa o caminho mais curto entre todas as cidades, usando o algoritmo de Kruskal.

Algoritmo de Prim:

- Escolhemos uma cidade inicial, por exemplo, BHO.
- Inicialmente, temos uma árvore vazia.
- Procuramos a conexão mais curta da cidade inicial para outra cidade e a adicionamos à árvore.
- Procuramos a próxima conexão mais curta que conecta a árvore existente a uma nova cidade.
- Repetimos o processo até que todas as cidades estejam na árvore.

BHO -- RJO (439)
RJO -- SPO (429)
CTB - SPO (408)
CTB - POA (711)
BHO - BRA (716)

BRA - CBA (1133)
 BHO - SAL (1372)
 REC - SAL (897)
 NAT - REC (298)
 FOR - NAT (537)
 CBA - MAN (2357)

Após adicionar todas as conexões necessárias, obtemos o grafo de conexão mínima pelo algoritmo de Prim.

O grafo fica assim para os dois métodos :

