

Copa do mundo no Brasil.

A copa do mundo sediada no Brasil em 2014, contou com 12 cidade sedes dos jogos. Marcos é um brasileiro que mora na Nigéria, ele ama futebol e decidiu assistir todos os jogos da Nigéria durante a fase de grupo. Porém Marcos teve problemas, o primeiro jogo que ele gostaria de assistir iria acontecer na cidade de Curitiba e a cidade não tem aeroporto internacional. Então Marcos deve desembarcar em algum aeroporto internacional (São Paulo, Rio de Janeiro, Brasília ou Natal) para depois conseguir ir até Curitiba. Marcos então decide desembarcar em São Paulo.

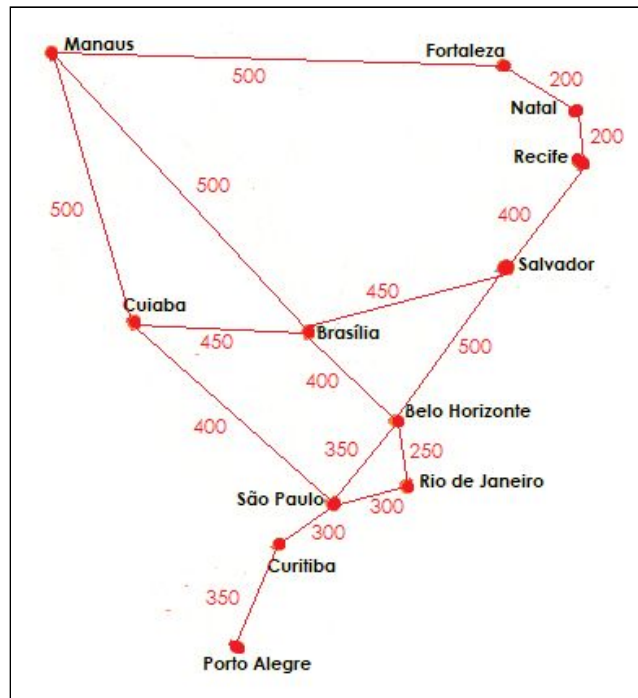
Por causa do grande volumes de voos nacionais, a ANAC (Agência Nacional de Aviação Civil) determinou que os aviões que trafegavam entre as cidades sedes dos jogos só poderiam ficar entre duas cidades específicas, ou seja, se um avião sai de São Paulo rumo a Curitiba, ele só poderá trafegar entre essas duas cidades indo e voltando. O mapa das rotas aéreas nacionais entre as cidades sedes dos jogos da copa com seus respectivos preços da passagem é representado abaixo:



As próximas cidades visitadas por Marcos para assistir os jogos eram: Porto Alegre, Cuiabá e Brasília, respectivamente. Antes de ir embora para Nigéria, Marcos desejava visitar sua família em Fortaleza e por fim ir ao aeroporto internacional de São Paulo.

Para conseguir resolver esse problema e fazer com que Marcos gaste o mínimo possível com as passagens aéreas, representamos o problema na forma de

Grafos. Grafos são compostos por vértices (representados pelas cidades sedes dos jogos) e arestas (representadas pelos voos entre às cidades sedes). Além disso, as arestas possuem pesos, que são representados pelos preços das passagens de cada voo e todas elas são bidirecionais. Diante disso, o Grafo que ilustra o problema é apresentado na figura a seguir:



Para descobrir qual o caminho que ele terá que fazer para gastar o mínimo possível, utilizamos o algoritmo de Dijkstra, onde dadas as informações de todos os voos e os pontos de origem e destino, o algoritmo informará qual o caminho mais barato. Às entradas no algoritmo são representadas nas colunas 1 e 2 da tabela a seguir. Na coluna 3 é retornado qual o caminho mais barato entre as entradas informadas. Na coluna 4 é apresentado o preço do caminho.

De	Para	Caminho	Preço
São Paulo	Curitiba	São Paulo -> Curitiba	300
Curitiba	Porto Alegre	Curitiba -> Porto Alegre	350
Porto Alegre	Cuiabá	Porto Alegre -> Curitiba -> São Paulo -> Cuiabá	1050
Cuiabá	Brasília	Cuiabá -> Brasília	450
Brasília	Fortaleza	Brasília -> Manaus -> Fortaleza	1000
Fortaleza	São Paulo	Fortaleza -> Manaus -> Cuiabá -> São Paulo	1400
Total gasto com passagens:			4550

Loja de eletrodomésticos.

O gerente de uma das maiores lojas de eletrodomésticos do Brasil decidiu comemorar o aniversário da loja de um jeito diferente. Através de um sorteio, um participante teria um carrinho de compras e 5 minutos para pegar qualquer item da loja. Marília se inscreveu e foi a sorteada. Quando foi chamada para participar, ela recebeu um carrinho de compras que suportava um peso de 300kg e teve a sua disposição todos os itens da loja. Marília deseja selecionar os itens mais valiosos da loja de forma que a soma dos pesos de todos eles não ultrapasse a capacidade máxima do carrinho. A tabela a seguir está representando o peso e valor dos itens mais valiosos disponíveis na loja:

Item	Valor (R\$)	Peso (Kg)
Geladeira Frost Free Inverse	2500	100
Máquina de lavar roupa 15 Kg	1800	80
Fogão 5 bocas / Elétrico	1000	40
Freezer Horizontal / 534 L	800	120
Micro-ondas 32 L / inox / espelhado	600	10
Depurador 60 cm / parede / inox	400	50

Para que Marília possa obter o maior lucro entre os itens disponíveis, podemos recorrer a combinatória e testar todas as combinações possíveis até encontrar a melhor delas. Entretanto, Marília tem apenas 5 minutos para testar todas as combinações possíveis e ir atrás dos itens, isso seria inviável. Essa alternativa apresentada é conhecida na técnica de projetos de algoritmos como *Backtracking*. *Backtracking* em muitos casos é muito útil, pois ele testa todas as combinações possíveis até encontrar a solução, mas em casos como esse que o tempo é limitado, essa técnica não é a melhor.

Para isso, temos outra técnica de projeto de algoritmos que é a Programação Dinâmica (PD). PD tem como principal objetivo obter a melhor solução (solução ótima) visando o melhor desempenho do algoritmo e por sua vez, levar menos tempo para encontrar a solução ótima. No problema apresentado acima, podemos dividir as informações em dois conjuntos (peso e valor) e submetê-lo ao algoritmo do problema da mochila.

valor [] =	2500	1800	1000	800	600	400
-------------	------	------	------	-----	-----	-----

peso [] =	100	80	40	120	10	50
------------	-----	----	----	-----	----	----

Com isso, Marília terá às informações à tempo de quais itens ela deverá pegar para obter o maior lucro. Para esse caso, a solução ótima é apresentada na tabela a seguir:

						Total
Itens	Depurador	Micro-Ondas	Fogão	Máquina	Geladeira	5 itens
Peso	50	10	40	80	100	280 Kg
Valor	400	600	1000	1800	2500	R\$ 6300

Algoritmos

Ambos os exemplos abordados anteriormente tem seus respectivos códigos disponíveis em:

Copa do mundo no Brasil

<https://github.com/freddomingues/ProgramacaoDinamica/blob/master/Grafo/src/CopaDoMundo.java>

Loja de Eletrodomésticos

<https://github.com/freddomingues/ProgramacaoDinamica/blob/master/ProgramacaoDinamica/src/mochila/LojaEletrodomesticos.java>