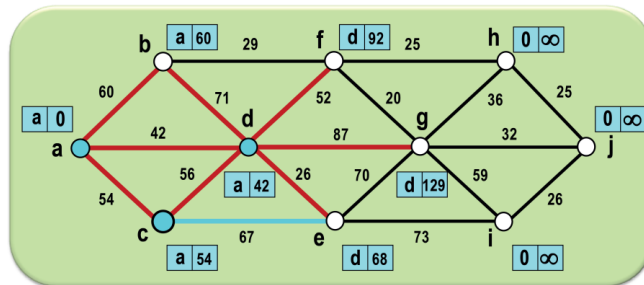
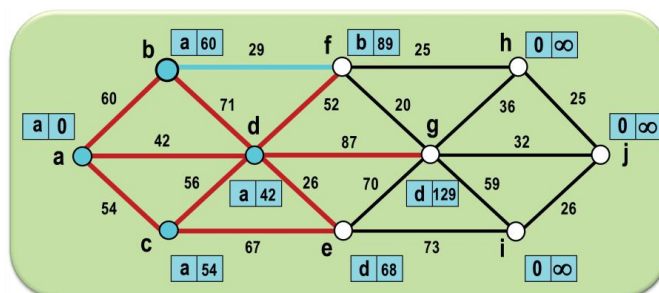


Verifica-se que houve um armazenamento do valor das arestas nos seus correspondentes vértices. Esses valores serão guardados para fins de comparação. Para entendimento do algoritmo neste caso, vamos fazer uma análise. Guarde o peso dos valores armazenados no vértice B,C e D, cujo respetivamente são: B(A,60), C(A,54) e D(A,42).

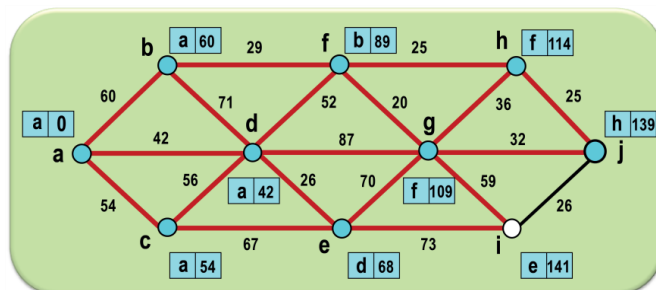


Nesta fase, devido aos valores coletados até o momento, o algoritmo define que a melhor rota passará pelo vértice D. Então vamos para o próximo vértice. No entanto, para evitar erros de escolha precipitada, ele compara com vértices anteriores para ver se D é realmente a melhor rota. Como pode-se ver na imagem acima, atualmente vai para a segunda opção mais viável além do vértice D, que neste caso seria o vértice C com peso 54. Para fazer uma correção, a rota **A-D-F** tem um valor de 94 em vez de 92, conforme mostrado na imagem acima, para melhor entendimento da imagem.

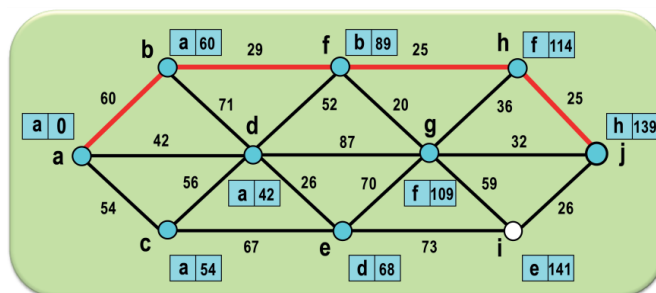


Assim como no caso dito anteriormente, ele fará a verificação a partir do vértice B, e olhe só o resultado. Aparentemente a **trajetória A-B-F** teve como peso total 89, e isso é bem menos do que qualquer outra rota traçada até então, tornando por enquanto está a melhor rota.

Seguindo essa lógica, ele fará para todos os próximos vértices e rotas que ainda faltam serem verificadas. Observe na próxima imagem o resultado do nosso algoritmo já avançado e chegando perto da sua conclusão final.



Nesse ponto, podemos observar a última comparação, que foi entre a **rota A-B-F-H-J de peso 139** com a **rota A-C-E-I-J de peso 167** ou que a **rota A-D-G-J que apresenta peso 141**. Concluindo que a primeira rota citada é a de menor custo, obtemos assim o resultado final.



Assim, o algoritmo de Dijkstra trabalha no grafo dado na questão. Somando os pesos e os comparando em cada vértice, a fim de traçar a menor rota possível e obter o melhor resultado.

Em nossos códigos usamos os mesmos vértices e pesos deste trabalho para replicar o resultado que acabamos de explicar, afim de comprovar na teoria e na prática o cálculo de menor caminho Dijkstra.

Resultado do nosso algoritmo Dijkstra implementado em c:

```
PS C:\Users\eliev\Documents\Git\universidade\Estruturas de dados2\disjkstra\dijkstra> .\a.exe
0: 1(60.00), 2(54.00), 3(42.00),
1: 5(29.00), 3(71.00),
2: 3(56.00), 4(67.00),
3: 5(52.00), 4(26.00), 6(87.00),
4: 6(70.00), 8(73.00),
5: 7(25.00), 6(20.00),
6: 7(36.00), 8(59.00), 9(32.00),
7: 9(25.00),
8: 9(26.00),
9:

Busca menor caminho
Id ant  peso
1: 0 -> 60.000000
2: 0 -> 54.000000
3: 0 -> 42.000000
4: 3 -> 68.000000
5: 1 -> 89.000000
6: 5 -> 109.000000
7: 5 -> 114.000000
8: 4 -> 141.000000
9: 7 -> 139.000000
PS C:\Users\eliev\Documents\Git\universidade\Estruturas de dados2\disjkstra\dijkstra> █
```

O código está zipado junto com este documento.