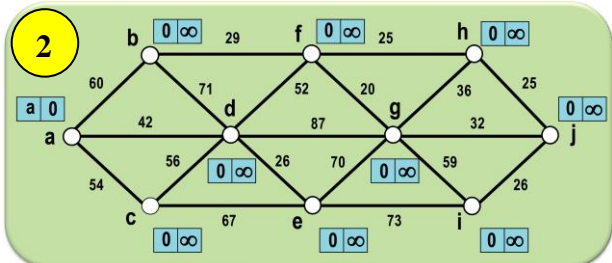
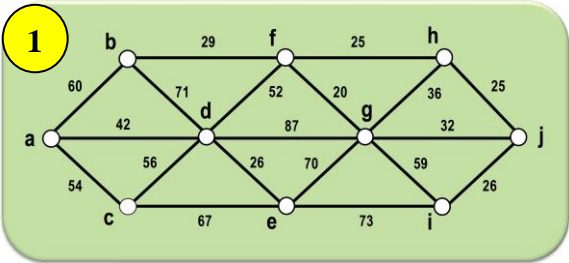
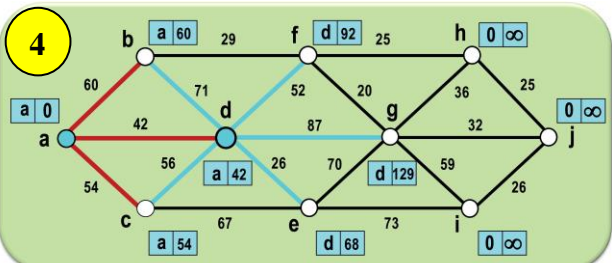
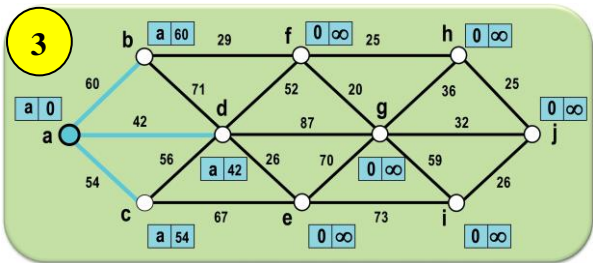


TRABALHO REFERENTE À
AVALIAÇÃO 3

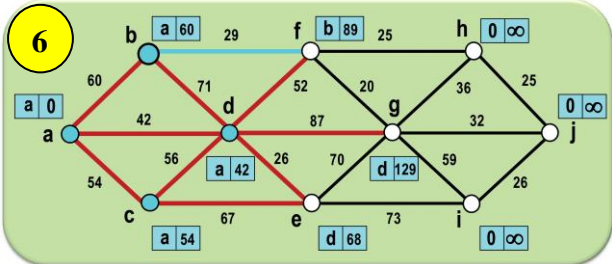
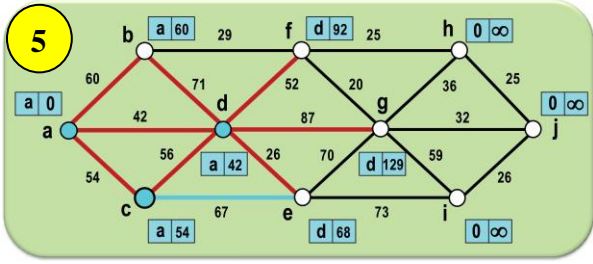
1) Explique o algoritmo de Dijkstra para o caminho mais curto em grafo usando as figuras do exemplo a seguir, depois implemente em C o algoritmo de Dijkstra.



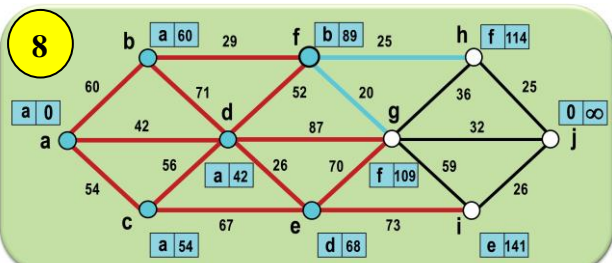
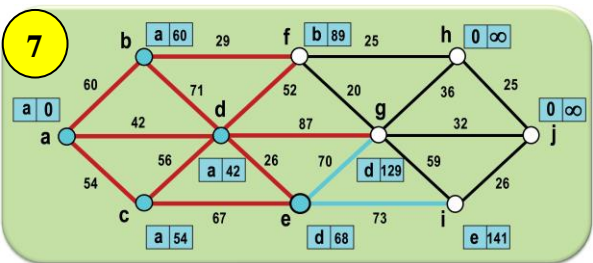
No algoritmo de Dijkstra, para chegar do vértice *a* para o vértice *j*, primeiramente, o vértice *a* verifica quais os vértices ele consegue alcançar diretamente, após isso, o algoritmo guarda a distância dos três percursos possíveis, de (*a*, *b*), (*a*, *d*) e (*a*, *c*), como podemos ver na imagem 3.



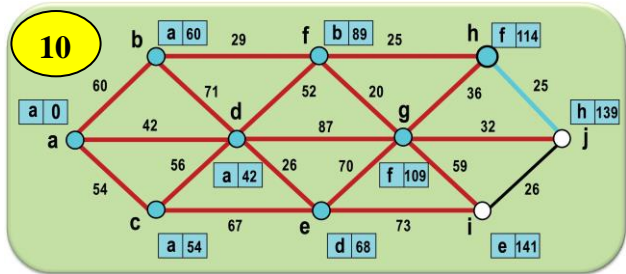
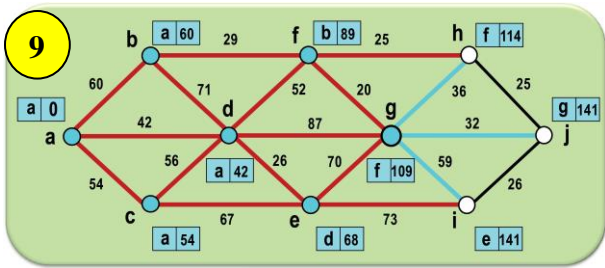
Agora, o algoritmo analisa para cada vértice alcançado pelo vértice inicial, sendo o *b*, *d* e *c*, quais novos vértices eles conseguem alcançar, começando pelo que tem menor distância, na imagem 4, o algoritmo verifica os vértices alcançados pelo vértice *d* e soma o valor da distância que ele já tem com o valor para estes novos vértices, no caso de vértices que já possuem um valor de distância definido, se essa soma tiver um valor menor que a distância desse vértice, esse valor é substituído. Na imagem 5, o algoritmo verifica se há algum vértice que o *c* alcança em que o valor da distância do vértice *c* com a soma da distância da aresta até um determinado vértice é menor que a distância que aquele vértice alcançado já possui, neste caso essa soma é sempre maior, então não há alteração.



Na imagem 6, o algoritmo verifica se algum vértice alcançado pelo vértice *b*, tem o valor da soma da distância das arestas menor do que o vértice alcançado já possui, como tem, o valor da distância do início até o vértice *f* é alterado. Agora, o algoritmo analisa os vértices *f*, *g*, e *e*, começando pelo que tem menor valor de distância para o vértice inicial, na imagem 7, é verificado se algum vértice alcançado pelo vértice *e* é novo ou se o valor da soma da distância até esse vértice alcançado é menor do que ele já possui, neste caso, o vértice *i* guarda o valor dessa soma. Na imagem 8, é verificado a mesma coisa da imagem 7, só que com o vértice *f*, neste caso, a distância para se chegar no vértice *g* recebe um novo valor, e a distância do início para o vértice *h* recebe o valor da distância de *f* somado com a distância de *f* para *h*.



Na imagem 9, o algoritmo de Dijkstra verifica se algum vértice alcançado pelo vértice **g** não possui valor de distância até ele ou se o valor da soma da distância até esse vértice alcançado é menor do que ele já possui, neste caso, o vértice **j** recebe o valor da soma da distância de **g**, somado com a distância de **g** para **j**.



Agora, o algoritmo analisa os vértices **h** e **i**, começando pelo que tem menor valor de distância para o vértice inicial, na imagem 10, é verificado se algum vértice alcançado pelo vértice **h** é novo ou se o valor da soma da distância até esse vértice alcançado é menor do que ele já possui, neste caso, como essa soma é menor, o vértice **j**, recebe essa nova distância. Na imagem 11, é verificado a mesma coisa da imagem anterior só que com o vértice **i**, mas como nenhuma das condições são satisfeitas, não ocorre nenhuma mudança de valores. Assim na imagem 12, mostra o resultado final do algoritmo de Dijkstra, onde é mostrado o menor caminho do vértice **a** ao vértice **j**, onde as arestas que, somando irão dar a menor distância estão marcadas.

