

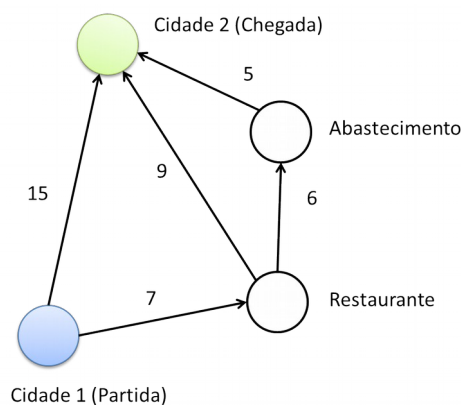
AEDII - 2022 (1s) - Exercício 14 - Bellman-Ford

Caminho Mínimo em Grafo com Restrição

Algoritmos e Estruturas de Dados II

Tempo limite: 1s

Em determinadas viagens muito longas, na maioria das vezes, torna-se indispensável realizar determinadas pausas para alimentação, abastecimento do veículo, descanso, entre outras coisas. Dessa forma nem sempre o caminho mais curto é o melhor caminho a ser seguido nesses trajetos. Isso pode ser observado na figura abaixo.



A partir disso, encontrar o caminho mínimo de um ponto até outro, exige em alguns casos o uso de restrições. Transpondo isso para um grafo, é possível afirmar, de forma simplificada, que na busca do caminho mínimo entre dois nós/vértices, deve-se considerar arestas com custo **menor ou igual** a um valor **R** (restrição).

Deve-se escrever um programa que exiba o caminho mínimo de um determinado **grafo orientado**, levando em consideração uma restrição nesse caminho mínimo encontrado, ou seja, o caminho mínimo que percorre arestas que tem, cada uma, custo menor ou igual a restrição informada. Essa operação deve ser realizado considerando apenas um nó de partida, em relação a todos os nós (vértices) do grafo. Por exemplo, se o grafo tem 4 nós/vértices, e o nó inicial for 2, então a saída vai ser composta por quatro números, que representam a distância (caminho mínimo com restrição) do vértice 2 para alcançar todos os demais nós do grafo.

Entrada

A entrada para cada teste se inicia informando o nó onde será iniciada a busca do caminho mínimo. O nó inicial tem valor **V** ($0 \leq V \leq 100$). O próximo valor a ser informado é o da restrição, representado por **R** ($-1024 < R < 1024$). Depois deve-se informar o número total de nós **N** ($0 \leq N \leq$

100) e o número de arestas **M** ($1 \leq M \leq 1024$). Nas próximas M linhas devem ser fornecidas as arestas que serão criadas juntamente com o peso delas, como, por exemplo, 2 7 -3 (possui uma aresta 2 -> 7 com peso -3). Nota-se que os pesos **P** das arestas estão no intervalo $-1024 < P < 1024$. Os nós são representados por valores numéricos que variam entre 0 e 100. Considerar a **ordem crescente** quando for inserido um novo nó na lista de adjacência. Por exemplo, considere que o nó/vértice 9 aponta (tem aresta de ligação) para 5, 6 e 8. Caso insira uma nova aresta 9 -> 3, o nó/vértice 3 deve ser inserido antes do nó/vértice 5 na lista de adjacência. Caso insira a aresta 9-> 4, então o nó/vértice 4 deve ser inserido antes do nó/vértice 5 e depois do nó/vértice 3 na lista de adjacência.

Saída

A saída do programa é exibida em uma linha que mostra o caminho mínimo com restrição do grafo com base em uma aresta inicial **V**.

Dicas:

- Podem-se utilizar as estruturas de dados (Pilha e Fila) desenvolvidas nos problemas anteriores;
- Não existem nós sem arestas, ou seja, isolados e não conectados (nós que não são ligados por nenhuma aresta);
- Arestas podem ter custo negativo;
- O custo do caminho mínimo que um nó leva para chegar a si mesmo é igual à zero.

Restrições:

- O programa deve ser escrito em C;
- Deve-se ter no código as funções de: inicialização, inserção e busca;
- Deve-se utilizar uma TAD Grafos Orientado por Lista de Adjacência;
- Não existem arestas com laços (arestas cujo vértice aponte para si próprio).

Exemplo de Entrada	Exemplo de Saída
0 8 8 9 0 1 4 0 3 -2 1 4 9	0 4 16 -2 14 9 17 21

1 5 5	
4 6 3	
6 0 -5	
5 2 7	
2 7 5	
2 4 -2	

Exemplo de Entrada	Exemplo de Saída
2	-73 -41 0 -19
0	
4 6	
1 0 -32	
1 2 28	
2 0 25	
0 3 -12	
3 1 -22	
2 3 -19	

Exemplo de Entrada	Exemplo de Saída
7	-5 -2 -7 2 -4 -6 3 0
3	
8 12	
1 0 -3	
2 0 3	
3 0 -2	
4 1 2	
6 3 -1	
5 2 1	
7 4 -4	
7 6 3	
7 5 6	
1 2 4	
4 2 -3	
2 5 1	