

**Aluna:** Ana Cláudia Machado

**Entrada:**

A entrada utilizada é um arquivo *.txt*, intitulado "graph", contendo as arestas e seus pesos. Assim, em cada linha do arquivo, têm-se, respectivamente, os vértices inicial e final daquela aresta e seu peso. Utiliza-se a biblioteca *networkx* para realizar a leitura das arestas e, em seguida, utiliza-se listas para armazenar e trabalhar com os vértices, as arestas e os pesos das arestas. A entrada utilizada para testes foi a proposta pelo professor.

**Funcionamento do algoritmo:**

Inicialmente, utiliza-se a estrutura dicionário para guardar as informações referentes a cada vértice: seu índice, o tempo mais cedo (EST), inicializado com zero, e o tempo mais tarde (LST), inicializado com um número suficientemente grande, no caso, 1000.

Em seguida, é iniciado o processo de encontrar o EST para todos os vértices, utilizando a lógica: dado a aresta  $(x, y)$ , ao constatar que  $y$  é o vértice final de uma aresta, confere-se se o EST de  $y$  é maior do que o EST de  $x$  somado com o peso de  $(x, y)$ . Caso contrário, EST de  $y$  possuirá, agora, o valor dessa soma.

Finalizado o processo de encontrar todos os tempos mais cedo, utiliza-se a ideia de uma fila de prioridades, na qual o primeiro vértice será aquele que não possui nenhum predecessor no grafo, ou seja, o final da última atividade realizada. Importante ressaltar que o LST desse vértice possui o mesmo valor de seu EST. A seguir, para cada vértice nessa fila, confere-se se tal vértice  $y$  é o final de alguma aresta. Caso sim, utilizando a mesma aresta  $(x, y)$  de exemplo, é calculado o LST para  $x$ , de modo que subtrai-se do LST de  $y$  o peso da aresta  $(x, y)$  e, se esse valor for menor do que o atual LST de  $x$ , então substitui-o pela soma explicitada acima. Além disso, se  $x$  ainda não estiver na fila de prioridades, adiciona-o. Esse procedimento é realizado para todos os vértices do grafo.

Por fim, para encontrar as arestas que fazem parte do caminho crítico, basta conferir se, para uma aresta  $(x, y)$ , se o LST de  $y$  subtraído do EST de  $x$  é igual ao peso da aresta. Em outras palavras, se a folga existente para aquela atividade é nula. Para as arestas que essa conferência for verdadeira, adiciona-as no caminho crítico.

**Saída:**

A saída é composta por 2 partes: primeiro, têm-se uma impressão do dicionário utilizado, no qual é possível visualizar os tempos mais cedo e mais tarde para cada vértice de acordo com seu índice. Por fim, a segunda parte é composta pelas arestas que fazem parte do caminho crítico naquele grafo. Para o grafo utilizado como teste na atividade, têm-se a saída:

```
machadoana@Ana:~/Grafos/PERT-CPM$ python3 pert-cpm.py
{'vertice': 1, 'EST': 0.0, 'LST': 0.0}
{'vertice': 2, 'EST': 7.0, 'LST': 7.0}
{'vertice': 3, 'EST': 5.0, 'LST': 12.0}
{'vertice': 4, 'EST': 16.0, 'LST': 16.0}
{'vertice': 5, 'EST': 16.0, 'LST': 18.0}
{'vertice': 6, 'EST': 27.0, 'LST': 27.0}
{'vertice': 7, 'EST': 20.0, 'LST': 22.0}
{'vertice': 8, 'EST': 30.0, 'LST': 30.0}
{'vertice': 9, 'EST': 36.0, 'LST': 36.0}
{'vertice': 10, 'EST': 34.0, 'LST': 36.0}
{'vertice': 11, 'EST': 43.0, 'LST': 43.0}

Faz parte do caminho critico as arestas:[(1, 2), (2, 4), (4, 6), (6, 8), (8, 9), (9, 11)]
machadoana@Ana:~/Grafos/PERT-CPM$ |
```