

Trabalhos Práticos da 2ª VA

Parte 1

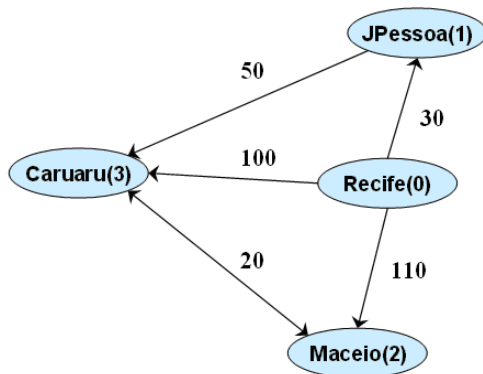
Aprimorar sua estrutura de dados criada para a 1ª VA para lidar com **grafos direcionados com pesos** nas arestas (*grafos ponderados*). (Grafos não-direcionados podem ser representados como grafos simétricos).

Requisitos:

- Usar representação de listas de adjacências.
- **Acrescentar**, pelo menos, as seguintes **funcionalidades**:
 - Função para **adicionar** no grafo uma **aresta (x,y) com peso p**, onde **x** e **y** são vértices válidos (inteiros de 0 a $n-1$) e **p** é um valor inteiro
 - Uma maneira de acessar (apenas) os sucessores de um nó **v** (entrada), junto com o peso da aresta até o sucessor
 - Uma função que retorne o **peso de uma aresta (x,y)**, para **x** e **y** dados como entradas, ou retorne um erro se não existir aresta
 - Uma função que retorne o **nome** de um dado vértice **v**.
 - Operação para ler um *grafo com pesos* de um arquivo texto conforme o novo formato abaixo

Novo formato do arquivo:

- A primeira linha indica a quantidade de nós/vértices **N**
- Cada linha seguinte se refere a um nó específico, na ordem de 0 a $N-1$
 - A primeira linha se refere ao nó 0, a segunda se refere ao nó 1, etc.
- Cada linha referente a um nó **v**, começa com uma string sem espaço, que representa o **nome** (ou rótulo) do nó **v**
- No restante da linha referente a um nó **v**, é dada uma lista de nós intercalados com pesos, representados como pares **u_i p_i** , sendo **u_i** um sucessor de **v** (ou seja, existe a aresta **$v \rightarrow u_i$**), sendo eles ligados por um arco de peso **p_i** .
- Exemplo (com o número do nó entre parênteses):



(representação visual)

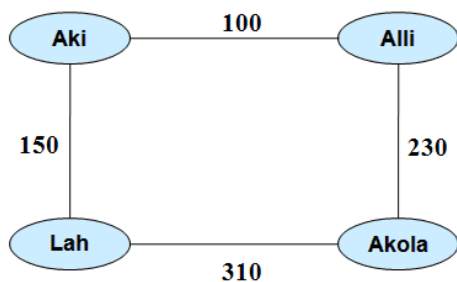
```
4
Recife 1 30 3 100 2 110
JPessoa 3 50
Maceio 3 20
Caruaru 2 20
```

(arquivo)

Parte 2

Certo país possui várias regiões (ilhas fluviais e margens) interligadas por pontes. Durante uma guerra, todas as pontes foram destruídas, deixando as diversas regiões isoladas. Por falta de recursos, o governo deseja reconstruir apenas o mínimo de pontes de modo a reintegrar todas as regiões do país.

Seu programa deve receber um **grafo não-direcionado** (representado como um direcionado simétrico) onde os vértices são as regiões e as arestas são as pontes que o país possuía antes da guerra. Considere também que o peso de cada aresta representa o custo de reconstruir a ponte. Um exemplo de entrada é mostrado abaixo.



(representação visual)

4				
Aki	1	100	2	150
Alli	0	100	3	230
Lah	0	150	3	310
Akola	1	230	2	310

(arquivo)

A saída deve **listar as pontes a serem reconstruídas**, bem como o **custo total mínimo** da reconstrução. Assim, para a entrada mostrada antes, seu programa deverá produzir a seguinte saída:

```
Pontes a serem reconstruídas:
    (Aki, Alli)
    (Aki, Lah)
    (Alli, Akola)
Custo total: 480
```

Parte 3

Seu programa deve receber duas entradas:

1. um **grafo direcionado** ou **não-direcionado** (simétrico), representando cidades (vértices) e estradas entre as cidades (arestas) com as respectivas distâncias (pesos das arestas),
2. e uma **cidade** inicial.

Você deve **achar todos os menores caminhos** para ir da cidade inicial para todas as outras. (Observe que a cidade inicial é fixa).

O programa dever dar como **saídas** todos os menores caminhos começando da cidade inicial (e terminando em todas as demais cidades).

Abaixo, segue a saída correspondente ao grafo dado na página 1, assumindo a cidade de Recife como cidade inicial:

```
Menores caminhos saindo de Recife:  
Recife->JPessoa (custo: 30)  
Recife->JPessoa->Caruaru->Maceio (custo: 100)  
Recife->JPessoa->Caruaru (custo: 80)
```