

MC346 - PROJETO 3 - V.1

Data de entrega: até 29/04

Dado um grafo, um vértice de saída e um de chegada imprimir o caminho mais curto e o mais rápido entre a saída e a chegada. outros vértices

O arquivo de entrada (pelo stdin) contém:

- um linha com o vértice saída e o vértice chegada, da forma

`s z`

que indica que os caminhos devem ser calculados saindo de `s` e indo para `z`.

- uma lista de arcos na forma

`a b 5.3 30.0`

um por linha, que indica que há uma rua os vértices `a` e `b` e o distancia deste arco é 5.3km, e a velocidade máxima neste trecho é de 30.0 km/h. Ruas são direcionadas, assim um arco entre `a` e `b` indica que da para ir de `a` para `b` por esta rua, mas não de `b` para `a`. Se a rua tem os dois sentidos, apareceria no arquivo tanto uma linha

`a b 5.3 30.0`

como uma linha

`b a 5.3 30.0`

Escreva um programa em haskell que lê do stdin o conjunto de ruas e o vértice de saída e o de chegada, e imprime o caminho com menor comprimento entre o vértice de saída e o de chegada, e o caminho que demoraria menos tempo (se você andasse na velocidade máxima em cada rua). Use o algoritmo de Dijkstra.

A saída de vera ser no formato:

`caminho mais curto`

```
s
b
c
z
13.4
```

`caminho mais rapido`

```
s
a
d
c
z
112
```

A primeira saída indica o vértice de saída s , e o caminho (para b , depois c) até o vértice de chegada. O número que se segue é a distância percorrida no caminho (em quilômetros).

A segunda saída indica o caminho mais rápido, e o número que se segue é o tempo em **minutos** (inteiro, **truncado**) para percorrê-lo. Note que os dados estão em km e km/h. Faça os cálculos do tempo em Float e só trunque o resultado na hora da impressão.

Há uma linha em branco entre as duas saídas.

O susy rodará seu programa como:

```
runghc prog3.hs < arqx.in
```

ou seja, o programa deve se chamar `prog3.hs`, o nome da função principal do seu programa deve ser `main` e a leitura é pelo `stdin` (`getLine` ou `getContents`)