12/06/2018

Programação I
Segundo Trabalho

Prof.: Flávio Miguel Varejão

Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho).

Considere a localização de *n* cidades numeradas seqüencialmente de 1 a *n* no estado do ES tal como no esboço de mapa apresentado mais abaixo. Os nomes das cidades e as suas respectivas coordenadas *x* e *y* estão armazenadas no arquivo **nome-coord.txt**. Cada estrada entre duas cidades *i* e *j* possui um custo por quilômetro (c_{ij}) que deve ser multiplicado pela distância (d_{ij}) para calcular o custo de viagem (CV_{ij}) entre as cidades. Os valores dos custos por quilômetro estão armazenados no arquivo **custos.txt**. Assim, para calcular o custo de viagem *CV* entre as cidades *i* e *j*, deve-se utilizar a seguinte expressão:

$$CV_{ij} = c_{ij} \, x \, d_{ij}$$

Considere que existem estradas entre todas as cidades, mas que nem sempre a estrada que vai da cidade *i* até a cidade *j* é a mesma da cidade *j* até a cidade *i*. Ou seja, o valor de c_{ij} pode ser diferente de c_{ji}.

Uma empresa distribuidora de produtos alimentícios pretende atuar nesse mercado. Para isso, deseja decidir onde deve construir seu centro de distribuição assim como obter informações sobre os percursos a serem percorridos por seus caminhões para atender as demandas de distribuição de produtos.

Faça um programa em Haskell para:

1. Ler o arquivo de dados **nome-coord.txt**.
2. Ler o arquivo de dados **custos.txt**.
3. Determinar a cidade que abrigará o centro de distribuição. O centro de distribuição corresponde a cidade de mínima excentricidade, isto é, a cidade cujo maior custo de viagem *CV* entre a cidade e qualquer outra cidade é mínimo. Para calcular o custo de viagem entre uma cidade e outra você deverá utilizar o algoritmo de Djkistra (https://en.wikipedia.org/wiki/Dijkstra's_algorithm). Caso haja empate entre duas ou mais cidades, escolher como centro de distribuição, a cidade de menor soma de coordenadas. Em caso de novo empate, escolher a cidade cuja primeira coordenada é a menor.
4. Determinar o “menor percurso” para que um caminhão saia do centro de distribuição, percorra todas as demais cidades uma a uma, sem repetição, e retorne ao centro de distribuição. Para determinar o “menor percurso” você deverá aplicar o método do vizinho mais próximo para o problema do caixeiro viajante (https://en.wikipedia.org/wiki/Nearest_neighbour_algorithm). Caso haja empate entre duas ou mais cidades, em qualquer ponto de decisão do algoritmo, escolher a cidade de menor soma de coordenadas. Em caso de novo empate, escolher a cidade cuja primeira coordenada é a menor.

5. Imprimir em um arquivo denominado **saida.txt** as seguintes linhas:

linha 1: nome da cidade do centro de distribuição definido no item 3.
linha 2: valor da excentricidade do centro de distribuição definido no item 3.
linha 3: sequência de cidades do percurso definido no item 4.
linha 4: valor do custo total do percurso definido no item 4.

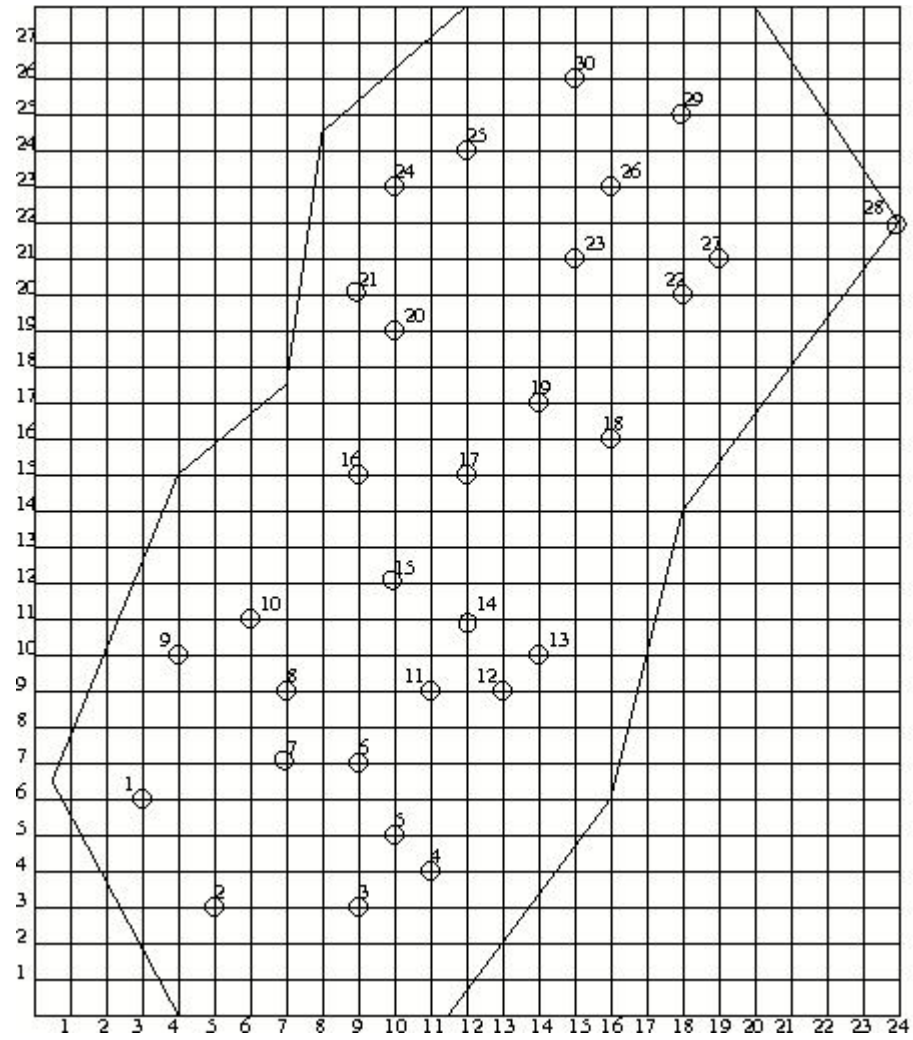
Observação:

a. Os arquivos de entrada **nome-coord.txt** e **custos.txt** e o arquivo de saída **saída.txt** devem estar no mesmo diretório do seu programa.

Apresenta-se, abaixo, um mapa ilustrativo de entrada de dados para o valor de *n* igual a 30. Os arquivos de entrada correspondentes **nome-coord.txt** e **custos.txt** são disponibilizados anexos a este enunciado.

V. Condições de Entrega

O trabalho deve ser submetido por e-mail até as 23:59 horas de 01 de julho de 2018 para o endereço josiasalexandre@gmail.com com o subject PROG1_TRABALHO_2_Nome_Completo_Aluno. O e-mail deve conter um arquivo .zip com o mesmo nome do subject do e-mail enviado contendo apenas o código fonte do trabalho. O arquivo principal (o que contém o main do trabalho) obrigatoriamente deve estar com o nome “main.hs”. Note que a data limite já leva em conta um dia adicional de tolerância para o caso de problemas de submissão via rede. Isso significa que o aluno deve submeter seu trabalho até no máximo um dia antes da data limite. Se o aluno resolver submeter o trabalho na data limite, estará fazendo isso assumindo o risco do trabalho ser cadastrado no sistema após o prazo.



Outras Observações Importantes:

1. O trabalho é INDIVIDUAL.
2. Trabalhos recebidos fora do prazo ou em formato inadequado recebem nota ZERO. Como dica, sugerimos não deixar para fazer o trabalho na última semana. Existe uma grande possibilidade de não conseguir cumprir os prazos.
3. Trabalho que não compila recebe nota ZERO. Não adianta nem apresentar.
4. Os trabalhos serão compilados e verificados usando o compilador ghc no sistema operacional Linux. Recomendamos que o desenvolvimento do trabalho seja feito neste ambiente.
5. Em momento algum, cabem contestações as regras estabelecidas nesta especificação de trabalho. Em outras palavras, se alguma dessas regras for violada, não adianta tentar qualquer argumentação pois sua nota NÃO será mudada.
6. Caso haja algum erro neste documento, serão divulgadas erratas. Portanto, fique atento às observações do professor durante as aulas e aos avisos na página do curso.

BOM TRABALHO!!!