TÉCNICO LISBOA

Programação

2021/2022

Enunciado do projecto (2º Período)

- O projecto deve ser realizado por um grupo de 4 elementos, devidamente criado para o efeito, no Fénix, até à data limite de 13-01-2022. Quem não respeitar a data limite será penalizado com 1 valor. Quem estiver com dificuldades na formação do grupo deve, até ao dia 14-01-2022, entrar em contacto com o Prof. Joaquim Dias para obter auxílio.
- Sem penalização por atraso na entrega, o projecto tem de ser submetido no Fénix até às 23:59 do dia 02-02-2022; com penalização de 1 valor, até às 00:59 do dia 03-02-2022; com penalização de 2 valores, até às 23:59 do dia 03-02-2022; com penalização de 4 valores, até às 23:59 do dia 05-02-2022. A partir daqui não se aceitam submissões.
- O projecto tem de ser entregue compactado num único ficheiro, forçosamente em formato ZIP (não se aceita, por exemplo, o formato RAR). O desrespeito desta indicação será penalizado com 1 valor. O ficheiro deverá ser chamado X.zip, onde X representa o nº do grupo.
- Eventuais indicações dos docentes a, de algum modo, autorizar o desrespeito do enunciado, só poderão ser invocadas na avaliação se tiverem sido publicadas na secção "Anúncios" da página web da cadeira.

Introdução

Considere o problema de determinar a localização de uma fábrica minimizando o custo total dos vários transportes associados ao seu funcionamento, e em particular, uma certa formulação simplificada deste problema – o "problema de Weber." Uma das formas de resolver este problema (bem como outros com a mesma formulação abstracta) é usando a heurística do centro de gravidade. O problema de Weber e a heurística do centro de gravidade são apresentados no Anexo 1.

O projecto consiste no desenvolvimento em MATLAB de uma aplicação de menu que permita a qualquer utilizador usar a heurística do centro de gravidade, de um modo agradável, na resolução do problema de Weber.

Para todos os grupos haverá uma reunião intercalar para apresentação do trabalho entretanto desenvolvido. Em certos casos haverá também uma discussão oral de todo o projecto. Os propósitos da reunião intercalar, bem como as possíveis situações de discussão oral, são esclarecidos no Anexo 2.

O projecto será classificado de um modo tão objectivo quanto possível. No Anexo 3 são apresentados os critérios mais importantes para a sua classificação.

Enunciado

Desenvolva em MATLAB uma aplicação de menu, chamada `localizacao`, para resolver o problema de Weber usando a heurística do centro de gravidade apresentada no Anexo 1. Para além do que se indica no cabeçalho deste enunciado, o projecto deve satisfazer dois tipos de requisitos: os mínimos, e os complementares. O incumprimento de requisitos mínimos é considerado muito grave. Assim, a menos de situações muito especiais que possam ser consideradas atendíveis, se um projecto não cumprir todos os requisitos mínimos sofrerá, só por isso, uma penalização de 4 valores.

Se o desejar, para além de uma versão do projecto que respeite os requisitos anteriores, versão essa que terá sempre de ser entregue, e sobre a qual incidirá o essencial da avaliação, pode ainda entregar uma versão melhorada do projecto. Esta segunda versão, cuja configuração terá de ser previamente autorizada por um docente, incluirá requisitos de um terceiro tipo – os requisitos extra – podendo, eventualmente, desrespeitar alguns dos requisitos da primeira versão, e servirá apenas para avaliar a obtenção de uma bonificação a acrescentar à avaliação da primeira versão.

Os requisitos a considerar são apresentados a seguir.

Requisitos funcionais mínimos `(rfm1)`

Usando um menu, o programa deve oferecer ao utilizador as seguintes funcionalidades:

- Carregar dados
- Efectuar cálculos
- Mostrar a localização
- Gravar a localização
- Sair do programa

Requisitos de implementação mínimos `(rim1 a rim18)`

O programa tem de se chamar `localizacao`. `(rim1)`

Relativamente às opções do menu, o programa tem de:

- Carregar dados `(rim2 e rim3)`
    - Pedir ao utilizador o nome completo do ficheiro de texto simples onde estão os dados do problema a resolver. O programa tem de funcionar com qualquer nome completo correctamente estabelecido que se queira dar ao ficheiro. (Relembra-se que o nome completo de um ficheiro inclui – se este a tiver – a

sua extensão, e que um ficheiro de texto simples pode ter uma extensão qualquer, e pode inclusive não ter extensão nenhuma.)

    - Ler os dados do problema a partir do ficheiro escolhido. O programa tem de ser geral, isto é, tem de resolver qualquer caso correctamente estabelecido que se queira colocar no ficheiro. Exige-se a utilização do formato indicado no Anexo 1.

- Efectuar cálculos `(rim4)`
    - Executar a função `centro`, que, usando a heurística do centro de gravidade, efectua todos os cálculos necessários para as duas opções seguintes. Naturalmente, a função `centro` tem de ser criada como parte do programa.
- Mostrar a localização `(rim5)`
    - Mostrar na janela de comandos as coordenadas da localização calculada. O que é escrito tem de se seguir o formato ilustrado. Por exemplo:

      ```
      Resolução do Problema de Localização
      Coordenadas: X = 500.45, Y = 120.52
      ```
- Gravar a localização `(rim6)`
    - Gravar para um ficheiro de texto simples exactamente a mesma informação que é mostrada na opção anterior. O nome do ficheiro tem de ser formado a partir do nome do ficheiro de entrada alterando a extensão para `.res`. Por exemplo: se o nome do ficheiro de entrada for `casoA.dat`, o nome do ficheiro de saída será `casoA.res`; se for `casoB` (sem extensão), o ficheiro de saída será `casoB.res` (assume-se aqui que o ficheiro de entrada nunca terá a extensão `.res`). Chama-se a atenção para o facto de que não faz sentido condicionar a utilização deste botão à utilização do botão anterior.
- Sair do programa `(rim7)`
    - Apenas sair do programa.

O menu tem de ser implementado com a função pré-definida `menu`. `(rim8)`

Complementarmente, em todo o programa, está interdito o uso de: `(rim9 a rim18)`

- Quaisquer tipos de interface gráfica para além da função pré-definida `menu`.
- Submenus.
- *Scripts*.
- Funções pré-definidas não abordadas nas aulas ou no livro (3ª edição).
- As funções pré-definidas `load` e `save`.
- A função pré-definida `diary`.
- Funções locais (subfunções).
- Eco.
- Variáveis globais, variáveis persistentes, ou funções embebidas.
- Recursividade.

Requisitos de apresentação mínimos `(ram1)`

A todos os alunos é exigida a presença na reunião intercalar.

Requisitos de entrega mínimos `(rem1 a rem3)`

Para além dos aspectos indicados no cabeçalho do enunciado (penalizações por atraso na entrega, ou por entrega num formato que não o ZIP), na entrega do trabalho é necessário atender ao seguinte:

- O programa `localizacao` entregue tem de executar com sucesso o caso apresentado no exemplo do Anexo 1. (Todos os grupos têm a obrigação de, imediatamente após terem submetido o projecto no Fénix, o descarregar, e, com a versão descarregada, fazer todas as verificações que possam antecipar vir a ser feitas pelo corpo docente.)
- Para o corpo docente poder verificar o requisito anterior, têm também de ser entregues o ficheiro com os dados de entrada para o caso do Anexo 1 (deverá chamar-se `anexo1.txt`) e o correspondente ficheiro de saída produzido pelo programa. Naturalmente, estes ficheiros têm de estar de acordo com o que foi especificado.
- A "Tabela de Requisitos" que é fornecida com o enunciado tem de ser entregue devidamente preenchida. Esta tabela tem todos os requisitos, um por linha, identificados simplificadamente. Para cada requisito tem de ser indicado o quanto este foi satisfeito, e, eventualmente, deixada uma observação muito breve (observações mais elaboradas ficam para o relatório). A satisfação será indicada por um valor entre 0 (0%) e 1 (100%). Por exemplo:

```
Requisito   Satisfação  Observação

rfm1        1

rim1        1
...
ram1        X           Este requisito fica assim, sem valor.

rem1        1
...
ric1        1
ric1        1
ric3        1
ric4        1           Fizemos vários testes e nunca abortou.
ric5        0.8         Há algumas situações sem feedback.
ric6        1           Verificámos tudo o que nos ocorreu.
ric7        0.2         Só temos as funções Dijkstra e Caminhos.
ric8        1
ric9        0.5         Ligámos pouco aos comentários.
ric10       1

rec1        0.7         Só apresentamos o manual de utilização.

rfe1        0.5         Fizemos uma versão com submenus.
```

Requisitos de implementação complementares (ric1 a ric10)

Melhorar o requisito de implementação mínimo `rim5`, substituindo-o por: (ric1)

- Executar a função `escreve` para mostrar na janela de comandos todos os resultados dos cálculos efectuados, bem como os dados relativos ao problema em análise. Naturalmente, a função `escreve` tem de ser criada como parte do programa. O que é escrito tem de se seguir o formato ilustrado, nomeadamente: incluir o nome do caso, alinhar as barras verticais, e não usar a notação científica para escrever os números. Por exemplo:

```
Localização e custo total para o caso SOCAPOL
Coordenadas: X = 500.45, Y = 120.52
Custo total: CT = 340987.15
---
Pontos de interesse
(xi, yi)          | Qi        | Ci        | di
(300.0, 80.0)     | 2.0       | 50.0      | 301.093
(100.0, 200.0)    | 34.0      | 50.0      | 150.102
... (etc.)
```

  Nota: supõe-se neste exemplo que o ficheiro de dados se chama SOCAPOL, independentemente da extensão que eventualmente tenha.

Melhorar o requisito de implementação mínimo `rim6`, substituindo-o por: (ric2)

- Executar a função `escreve` (exactamente a mesma função que é usada na opção anterior) para gravar para um ficheiro de texto simples exactamente a mesma informação que é mostrada na opção anterior. O nome do ficheiro tem de ser formado a partir do nome do ficheiro de entrada acrescentando ao nome a informação de quantos pontos de interesse o caso em análise tem e mantendo a extensão. Por exemplo: se o nome do ficheiro de entrada for `casoA.dat`, e o caso tiver 7 pontos de interesse, o nome do ficheiro de saída será `casoA_7.dat`; semelhantemente, se o nome do ficheiro de entrada fosse `casoB`, o ficheiro de saída seria `casoB_7` (sem extensão).

Tal como todos os programas, este programa deve estar escrito de modo a:

- Dar ao utilizador uma experiência o mais agradável possível, seja pela simplicidade e clareza dos processos usados, seja pela consistência da interacção (não usando formas de interacção diferentes para situações semelhantes). (ric3)
- Garantir que o programa nunca aborta devido à existência de dados incorrectos ou devido a acções do utilizador (quaisquer que elas sejam). (ric4)
- Garantir que sempre que se escolhe uma opção no menu se tem algum *feedback* disso, mesmo que o utilizador tenha feito algo ilógico. (ric5)
- Garantir que os dados entrados estão correctos. (ric6)
- Respeitar os princípios da decomposição funcional. (ric7)
- Representar a informação que circula no programa de um modo adequado. (ric8)

- Usar um estilo de programação claro e respeitador das convenções comummente aceites na comunidade de programação. (ric9) Em particular, chama-se a atenção para:
    - Todas as funções devem ter um primeiro bloco de comentários escrito de acordo com a convenção habitual em Matlab.
    - O código deve estar comentado de um modo claro e conciso. Os comentários devem servir, sobretudo, para ajudar à compreensão do algoritmo ou para elucidar sobre a natureza dos dados; nunca devem ser usados para "ensinar" a programar. Por outras palavras, deve pressupor-se que quem vai ler o programa é um programador experiente, e os comentários devem servir para o ajudar a perceber o programa mais depressa.
    - As opções de estilo devem ser consistentes ao longo de todo o programa. Por exemplo, tendo-se optado por usar `uma_variavel`, não se deve depois usar `outraVariavel`.
    - Em geral, os identificadores devem ter nomes sugestivos.
    - Em geral, deve-se destacar os operadores. Algumas excepções: potenciação, operadores unários (e.g., transposição), e expressões muito curtas.
    - Os operadores de atribuição devem-se destacar sempre.
    - Na invocação das funções, nunca se deve deixar espaço entre o nome da função e o parêntese.
    - As listas de parâmetros ficam mais claras se se deixar um espaço a separar os parâmetros (a seguir a cada vírgula).
    - Em geral, os ciclos, incluindo as suas eventuais inicializações e pré-alocações, devem estar destacados. O mesmo se deve passar com as instruções de selecção.
    - Não se devem deixar, sistematicamente, linhas em branco entre instruções. As linhas em branco entre instruções devem servir para separar blocos de instruções com objectivos de algum modo separados.
    - Salvo situações muito excepcionais, o código não deve ultrapassar a linha vertical limite sugerida pelo editor de código. O mesmo se aplica aos comentários.
    - O código deve estar correctamente indentado. Para o efeito, basta usar o *Smart Indent* do editor de código (i.e., no editor, basta fazer: Ctrl+A, Ctrl+I).

Neste trabalho também se valoriza o seguinte requisito didáctico: (ric10)

- Por vezes o editor integrado do MATLAB apresenta sugestões ou "advertências" colocando o código com sublinhado ou com fundo colorido. Se o código entregue tiver situações destas, elas devem ser comentadas, na própria linha (ou na seguinte, se não couber na própria), a explicar porque ficaram assim.

Requisitos de entrega complementares (rec1)

Juntamente com os restantes elementos do projecto (o programa, os ficheiros de entrada e de saída, e a tabela de requisitos) deve ser entregue um pequeno relatório, com o máximo de 2500 palavras, em formato PDF. O ficheiro com o relatório deve chamar-se Relatório_X, onde X

representa o número do grupo. O relatório tem de incluir um manual de utilização do programa, usando para o efeito o exemplo apresentado no Anexo 1. O relatório poderá ainda servir para o grupo apresentar decisões tomadas no desenvolvimento do projecto que queira destacar, ou apresentar os testes que efectuou para garantir o cumprimento dos requisitos.

Requisitos funcionais extras `(rfe1)`

Do ponto de vista da avaliação, o principal objectivo destes requisitos é o de servir para compensar eventuais imperfeições no cumprimento dos restantes requisitos.

Os grupos são livres de sugerir quaisquer requisitos extra que entendam ser interessantes. Nenhum grupo está autorizado a implementar um requisito extra sem o discutir previamente com o corpo docente.

Por outro lado, quem decidir entregar um projecto com requisitos extra, terá de entregar duas versões do projecto: uma sem requisitos extra, e outra com requisitos extra. A segunda versão só será usada para avaliar os requisitos extra; todos os outros requisitos serão avaliados apenas na primeira versão. Se forem entregues duas versões, o ficheiro ZIP a entregar no Fénix deverá conter duas pastas, cada uma com a sua versão do projecto.

Deixa-se uma sugestão:

- Em alternativa ao menu descrito nas funcionalidades mínimas, a interacção com o utilizador pode estar organizada em submenus. Por exemplo, o menu inicial poderia conter apenas as opções “Carregar dados” e “Sair do programa”; o seguinte, as opções “Efectuar cálculos”, “Voltar atrás” e “Sair do programa”; e assim por adiante.

Anexo 1 – O Problema de Weber, a heurística do centro de gravidade, e a sua implementação no projecto

O problema de Weber

O problema de Weber – a versão que aqui nos interessa – pode ser enunciado do seguinte modo: Considere que se pretende construir as instalações industriais de uma empresa. As instalações serão o centro de deslocações, seja a fornecedores, seja a clientes, aqui genericamente designados por pontos de interesse. Para os i pontos de interesse, são conhecidas as suas coordenadas (xi, yi), as quantidades Qi a transportar em cada ligação a i, e os custos de transporte unitário Ci para cada ligação a i. Determine a localização das instalações industriais – as suas coordenadas (X, Y) – de modo a minimizar o custo total dos transportes CT = Σ Qi Ci di, onde di é a distância euclidiana das instalações ao ponto de interesse i.

A heurística do centro de gravidade

Uma das formas de resolver o problema antes proposto é usando a heurística do centro de gravidade que consiste simplesmente em aplicar as seguintes fórmulas: X = Σ xi Ci Qi / Σ Ci Qi ; Y = Σ yi Ci Qi / Σ Ci Qi.

Formato do ficheiro de dados

No projecto, os dados são obtidos a partir da leitura de um ficheiro. Exige-se que os dados estejam escritos no ficheiro com o seguinte formato:

```
(xi, yi),   Qi,   Ci
(300.0, 800.0), 2.0,  50.0
(800.0, 200.0), 3.0,  50.0
(200.0, 500.0), 2.5,  75.0
(600.0, 400.0), 1.0,  75.0
(800.0, 800.0), 1.5,  75.0
```

Nota: O ficheiro deve incluir o cabeçalho indicado, e os espaços em branco podem ser em qualquer quantidade ou até substituídos por *TABs*.

Utilização da função `centro`

No projecto, os cálculos têm de ser efectuados por uma função designada por `centro`. Exige-se que esta função tenha a seguinte assinatura:

```
[local, d, CT] = centro(coords, Q, C)
```

onde:

`coords` = matriz N x 2 com as coordenadas dos `N` pontos de interesse

`coords(i, 1)` = coordenada `x` do ponto de interesse `i`

`coords(i, 2)` = coordenada `y` do ponto de interesse `i`

`Q` = vector com as quantidades a transportar para os pontos de interesse

`Q(i)` = quantidade a transportar para o ponto de interesse `i`

`C` = vector com os custos de transporte unitário para os pontos de interesse

`C(i)` = custos de transporte unitário para o ponto de interesse `i`

`local` = vector com as coordenadas `X` e `Y` da localização das instalações

`local(1)` = coordenada `X` da localização

`local(2)` = coordenada `Y` da localização

`d` = vector com as distâncias euclidianas das instalações aos `N` pontos de interesse

`d(i)` = distância euclidiana das instalações ao ponto de interesse `i`

`CT` = custo total dos transportes

Caso de teste

A fim de se poder testar a heurística do centro de gravidade, deixa-se aqui um caso de teste com cinco pontos de interesse.

Dados de entrada:

(xi, yi): (300.0, 800.0), (800.0, 200.0), (200.0, 500.0), (600.0, 400.0), (800.0, 800.0)

Qi: 2.0, 3.0, 2.5, 1.0, 1.5

Ci: 50.0, 50.0, 75.0, 75.0, 75.0

Dados de saída (utilizando a heurística do centro de gravidade):

(X, Y): (516.0, 518.0)

di: 355.218, 426.357, 316.512, 144.845, 400.225

CT: 214710.03

Anexo 2 – Reunião intercalar e discussões orais

Reunião intercalar

O grupo deverá reunir com um docente da cadeira no período de 24-01-2022 a 26-01-2022. Nesta reunião intercalar, o grupo (completo) deverá apresentar o estado actual do trabalho, podendo então ser discutida a forma de melhorar o projecto. O agendamento desta reunião deve ser solicitado pessoalmente pelo grupo até ao dia 21-01-2022, caso contrário considera-se que o grupo não está interessado em fazer o projecto.

Todos os grupos têm de trazer consigo para a reunião intercalar (numa pen) o código que já tenham desenvolvido e uma versão provisória da tabela de requisitos a que o enunciado se refere, devidamente preenchida, mas apenas no que aos "Requisitos de implementação mínimos" diz respeito. Cada um destes requisitos deverá ter na coluna "Satisfação" o valor (entre 0 e 1) que o grupo considera adequado ao grau de desenvolvimento já alcançado.

Atenção: eventuais incumprimentos relacionados com o agendamento da reunião, ou com a ausência dos itens indicados no parágrafo anterior, serão penalizados com 1 valor cada.

Nesta reunião, se o trabalho entretanto desenvolvido for considerado pelo menos próximo do desejado (é desejado que, nesta fase, o projecto cumpra já todos os requisitos funcionais e de implementação mínimos), os alunos que demonstrem estar por dentro do que foi feito obterão 4 valores (ver o Anexo 3); os outros obterão uma cotação proporcional ao que mostrarem saber.

A propósito da reunião intercalar, se houver alguém no grupo que não esteja a participar no trabalho, nesta altura já os outros elementos do grupo o perceberam, e exige-se que esses casos sejam comunicados na reunião. Por outro lado, se algum elemento começar a não participar depois desta reunião, os restantes elementos do grupo devem comunicar o facto ao corpo docente o mais rapidamente que lhes for possível.

Discussões orais

Qualquer grupo pode ser chamado pelo corpo docente a apresentar o seu trabalho numa discussão oral. Numa tal situação, diferentes elementos do grupo poderão ficar com diferentes notas no projecto. Esta discussão oral ocorrerá durante o período de avaliação estabelecido pelo IST para o efeito, pelo que se recorda aos alunos que não se devem ausentar da vida académica neste período. Situações excepcionais, a existirem, terão de ser comunicadas atempadamente ao corpo docente, para haver ainda tempo de se encontrar uma alternativa.

Qualquer aluno a quem tenha sido aplicada a redução automática da nota referida no Anexo 3, e que entenda que a nota que tinha antes de fazer os exames reflete o seu conhecimento do projecto, pode pedir uma discussão oral individual para defender a nota anterior à redução.

Anexo 3 – Critérios de avaliação

Nota prévia: as indicações de valores que se seguem pressupõem uma avaliação do projecto na escala de 0 a 20 valores.

O incumprimento de requisitos mínimos é considerado muito grave. Assim, a menos de situações muito especiais que possam ser consideradas atendíveis, se um projecto não cumprir todos os requisitos mínimos, sofrerá, só por isso, uma penalização de 4 valores. Para além desta penalização, cada incumprimento de requisito mínimo dará origem a uma penalização de 2 valores, com excepção do requisito `rem1`, que será penalizado com 4 valores, do requisito `rem2`, que será penalizado com 1 valor, e do requisito `ram1`, que implicará a desistência do projecto por parte do(a) aluno(a) em causa.

Ainda a propósito de penalizações, recordam-se os aspectos indicados no cabeçalho deste enunciado que poderão dar origem a penalizações por atraso na entrega ou por entrega num formato não especificado (exige-se a utilização do formato ZIP).

Quanto aos restantes aspectos a considerar na avaliação, eles serão repartidos conforme segue:

- Reunião intercalar (4 valores). Esta parcela pode variar entre alunos do mesmo grupo.
- Requisitos de implementação complementares (14 valores), com a seguinte distribuição:
  - `ric1` (1.0 valor)
  - `ric2` (1.0 valor)
  - `ric3` (0.5 valores)
  - `ric4` (2.0 valores)
  - `ric5` (1.0 valores)
  - `ric6` (2.5 valores)
  - `ric7` (2.5 valores)
  - `ric8` (1.5 valor)
  - `ric9` (1.5 valores)
  - `ric10` (0.5 valores)
- Relatório (1.0 valor)
- Apreciação global (1.0 valor)
- Entrega de versão melhorada (com requisitos extra) (2 valores, no máximo)

Quando as notas de projecto de um grupo são obtidas sem discussão do projecto, qualquer elemento do grupo poderá pedir a redução da sua nota se achar que esta não corresponde ao seu efectivo envolvimento no resultado final. Por outro lado, a classificação obtida por um aluno num projecto sem discussão será automaticamente reduzida para 9.5 valores se o aluno obtiver num qualquer exame uma classificação não arredondada inferior em pelo menos 3 valores à nota do projecto (atenção, se o aluno tiver ido aos dois exames, para este efeito conta a menor nota). Para anular esta redução automática o aluno poderá solicitar a defesa da nota anterior à redução através de uma prova oral individual sobre o projecto.

Quando as notas de projecto de um grupo são obtidas a partir de uma discussão do projecto, as notas dos vários elementos do grupo já não poderão sofrer qualquer alteração.