

1 Objectivos

Utilização da linguagem Java e dos tipos abstratos de dados leccionados e implementados pelos alunos, para resolver um problema do tipo "Sopa de Letras". O aluno deve providenciar juntamente com os ficheiros pedidos a implementação de todas as estruturas de dados que usar na resolução do trabalho.

2 Descrição

Pretende-se que o seu programa seja capaz de produzir soluções para um jogo de Boggle. Este jogo aparece em alguns jornais americanos e rivaliza com as habituais palavras cruzadas ou as mais antigas sopas de letras. No Boggle há uma grelha de 4x4 letras, sendo o objectivo do jogo encontrar o maior número de palavras possível e quanto mais letras tiverem as palavras encontradas maior a pontuação. As palavras são formadas começando por uma qualquer letra da grelha e seguidamente uma qualquer letra adjacente. A única restrição é que não é possível repetir a mesma letra(posição), na mesma palavra.

A figura abaixo apresenta um exemplo real dum boogle do Miami Herald. As palavras "mouse", "monkey" e "mule" por exemplo, são possíveis de encontrar no tabuleiro de jogo apresentado, usando as regras descritas.



Figure 1: Exemplo do jogo

2.1 O trabalho

São fornecidos para a realização do trabalho dois ficheiros:

- um, com um boggle, isto é uma matriz de caracteres
- outro, uma lista de todas as palavras do Inglês, num ficheiro de texto, uma palavra por linha(um dicionário). Os boggles e o dicionário são em Inglês, dado que os caracteres característicos do português, tipo ç, ã, õ, etc, costumam trazer problemas adicionais que em nada se traduzem em conhecimentos de Estruturas de Dados e Algoritmos. Não se preocupe com as palavras do dicionário, ou com a sua destreza para encontrar palavras em Inglês, no boggle; esse é o trabalho do programa. Para todos os efeitos o seu programa consultará o dicionário, e se a palavra consta do dicionário, pode usá-la.

O output do trabalho, deverá ser uma lista com as palavras encontradas. Deverá providenciar o output da lista, devendo ser listadas as soluções encontradas na seguinte forma:

mouse (M:(2,1))->(O:(2,0))->(U:(1,1))->(S:(0,0))->(E:(0,1)). Para a realização do trabalho deverá implementar a classe **Position**, que agrupa no mesmo objecto o conceito de linha e coluna. Para esta classe deverá definir um construtor, e os métodos **toString** e **equals**. Não se esqueça deste último dado que o método **contains** para as listas usa o **equals**. A estratégia a seguir é simples, vão-se formando palavras usando uma posição do boggle e as suas posições adjacentes. Se a palavra encontrada é prefixo de uma palavra do dicionário, pode continuar guardando na lista de posições a nova posição. As posições adjacentes são N,S W,E, NE, SE, SW e NW. Tenha somente em atenção que na construção duma palavra não pode repetir posições. Se a palavra formada não é sufixo mas uma palavra do dicionário então guarda a palavra e a lista das posições na lista que contem os resultados.

3 Entrega do trabalho

O trabalho deverá ser entregue até ao dia 9 de Janeiro de 2020, sendo realizada a submissão pelo moodle, nos moldes habituais. Deve submeter os seguintes ficheiros:

- Uma implementação de Listas(que vai usar no trabalho)
- Uma implementação de Tabelas de Hash com acesso fechado, à sua escolha. Estas implementações deverão seguir as sugestões dadas nas aulas práticas(uma classe abstracta **HashTable** e uma subclasse com o **hasinhg** fechado de acesso que escolher). A tabela de hash servirá para guardar as palavras do dicionário e os seus prefixos
- A classe **Boggle**. Esta classe deverá ter os métodos que permitem resolver o problema proposto: **ler boogle(...)** e um método **solve()** que retorna a lista com as soluções encontradas
- Todas as classe que considerar necessárias para a construção da solução, incluindo a classe **Position**
- Um relatório adequado à apresentação do trabalho

O trabalho deve ser realizado em grupos de 2 elementos. Todos os ficheiros do trabalho deverão ser "zipados" e submetidos num único ficheiro com o nome **N1_N2_Boggle.zip**, para N1 e N2 os números dos alunos que realizaram o trabalho. Ambos os elementos do grupo submetem o trabalho, na sua área. O trabalho será apresentado presencialmente por ambos os elementos do grupo no dia 16 de Janeiro.