

1 Objectivos

Utilização das Estruturas de Dados leccionadas e da linguagem C para resolver um problema real do tipo "Sopa de Letras". O aluno deve providenciar juntamente com os ficheiros pedidos a implementação de todos os tipos 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 aparecia em alguns jornais americanos e rivalizava com as habituais palavras cruzadas ou as mais antigas sopa de letras. Hoje, também porque os jornais são digitais, é possível jogar on-line o jogo do Boggle mesmo sem ter de esperar pelo Boggle do jornal seguinte.

Regras do Jogo: No Boggle original, há uma grelha de 4x4 letras, sendo o objectivo do jogo encontrar as palavras escondidas na grelha. As palavras são formadas juntando caracteres adjacentes ao carácter escolhido. A única restrição aplicável na escolha dos caracteres usados para a formação das palavras, é que não podemos repetir a mesma posição.

Na figura 1 o exemplo da esquerda é uma edição dum boggle do Miami Herald. As palavras *mouse*, *monkey* e *mule*, são exemplos de palavras possíveis de encontrar neste boggle usando as regras do jogo.

No boggle da direita da figura 1, a palavra *BEFORE* está assinalada como uma das soluções deste boggle.

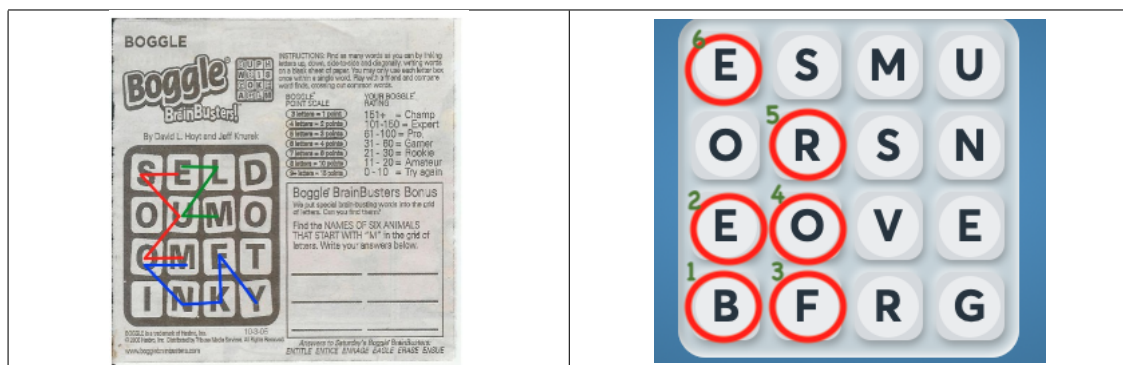


Figure 1: Exemplos de Boggles

3 Realização

São fornecidos para a realização do trabalho dois ficheiros:

- um, `boggle1.txt`, com um boggle, isto é uma matriz de caracteres
- outro, `cornicob_caps_2023.txt`, uma lista de todas as palavras no Inglês, num ficheiro de texto, uma palavra por linha.

Quer no boggle, quer na lista de palavras os caracteres são maiúsculos.

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 com conhecimentos de Estruturas de Dados e Algoritmos. Também não encontrei on-line nenhum jogo de boggle em português, logo os exemplos que serão tratados são em inglês. Não se preocupe se não dominar o idioma, interessa-lhe dominar as estruturas de dados e o processo de pesquisa das palavras do boggle. A existência da palavra que usa como solução é possível, se a palavra constar no dicionário. Não importa se sabe o seu significado, não lhe é pedida essa tarefa. Também o contrário se poderá verificar, i.e. uma palavra que pense que existe, se não estiver no dicionário não a poderá usar.

O output do trabalho, deverá ser a informação correspondente ao nº total de palavras encontradas, e a lista das palavras encontradas, bem como o percurso na grelha que permitiu a formação da palavra. A figura 2 apresenta a listagem de algumas das soluções do boggle (apresentado à esquerda) na figura 1.

```

ONE O:(2,0)->N:(3,1)->E:(2,2)
MOUSE M:(2,1)->O:(1,0)->U:(1,1)->S:(0,0)->E:(0,1)
MOULD M:(2,1)->O:(1,0)->U:(1,1)->L:(0,2)->D:(0,3)
MOO M:(2,1)->O:(1,0)->O:(2,0)
MOON M:(2,1)->O:(1,0)->O:(2,0)->N:(3,1)
MUSE M:(2,1)->U:(1,1)->S:(0,0)->E:(0,1)
MULE M:(2,1)->U:(1,1)->L:(0,2)->E:(0,1)
MULES M:(2,1)->U:(1,1)->L:(0,2)->E:(0,1)->S:(0,0)
MUM M:(2,1)->U:(1,1)->M:(1,2)
MOO M:(2,1)->O:(2,0)->O:(1,0)

```

Figure 2: Exemplo do formato das soluções

A estratégia a seguir para a construção das soluções é simples: vão-se formando palavras usando uma posição do boggle (linha e coluna da matriz) e uma das posições adjacentes. Se a palavra formada é prefixo de uma palavra do dicionário e a última posição considerada não é uma posição repetida, pode continuar, guardando a nova posição.

As posições adjacentes são *N, S, W, E, NE, SE, SW* e *NW*.

Se a palavra formada não é um prefixo mas uma palavra do dicionário então guarda a palavra e a lista das posições na lista que conterá as soluções. No fim, faz uma listagem das soluções.

Como saber se as palavras que vamos formando são prefixos de palavras existentes? Quando lê o ficheiro com a lista de palavras, vai, além de guardar a palavra lida numa tabela de hash, construir todos os prefixos possíveis dessa palavra e guarda-os noutra tabela de hash. Poder-se-á pensar que o número de prefixos das palavras é "intratável", mas de facto há muitos prefixos repetidos. Não se engane ao adicionar também, a palavra ao dicionário de prefixos. Terá uma tabela de palavras e uma outra de prefixos.

3.1 Entrega

O trabalho será realizado em grupos de 2 ou 3 elementos. A data limite para a submissão do trabalho é 25 de Junho, sendo realizada a submissão pelo moodle, nos moldes habituais. Só um elemento do grupo submete o trabalho.

O trabalho será acompanhado dum relatório, devendo a submissão ser constituída por todos os ficheiros usados para a elaboração do projecto, desde boggles a dicionários, etc. No main do seu projecto, não se esqueça de indicar os nomes dos ficheiros que usa para a execução do programa, sem "caminhos", i.e. nada de C:/Desktop/aulas..., etc. No meu computador os caminhos são outros.

Tudo na mesma directoria, para não haver erros de execução. Todos os ficheiros deverão ser "zipados" e submetidos num único ficheiro com o número dos alunos que realizam o trabalho e o nome do trabalho (ex: "NNNNN-CCCCC-TTTT-boggle.zip").

A apresentação dos trabalhos é obrigatória para todos os elementos do grupo e será realizada nos dias 26 e/ou 27 de Junho.