
Palavras Cruzadas

Objetivo: O objetivo deste trabalho é modelar um problema como um CSP e aplicar as técnicas estudadas na disciplina para resolvê-lo.

Administrativo: O trabalho deve obedecer as seguintes regras:

- Grupos: o trabalho pode ser feito em **duplas ou trios**, não sendo possível realizar individualmente.
- Data de Entrega: [14/10/2024](#) até às 23:59, via moodle. Apenas um dos integrantes deve submeter os arquivos do trabalho no moodle.
- Apresentações: Aulas dos dias [15/10/2024](#) e [17/10/2024](#). Todos os integrantes devem estar presentes e devem participar da apresentação.
- Fórum: Há um fórum no moodle onde devem ser indicados os grupos juntamente com o dia preferido para a apresentação. A escolha do horário de apresentação é por ordem de indicação no fórum.
- Avaliação: A nota deste trabalho será dividida em: implementação (80%) e apresentação (20%).

O problema

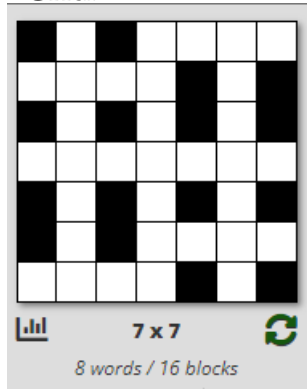
Nesse trabalho você deverá modelar o problema de preencher um grid de palavras cruzadas com um conjunto fornecido de palavras como um **Problema de Satisfação de Restrições**. Após isso, você pode empregar as técnicas estudadas em aula para resolver o problema proposto. Podem ser usadas combinações das técnicas estudadas, juntamente com outras ideias propostas pelo grupo para resolver o problema.

Devem ser respeitados os formatos de entrada e saída solicitados. Não é necessário implementar uma interface gráfica para o problema.

Entrada: um arquivo texto, com o formato fornecido pelo site Words Up? (<https://wordsup.co.uk/>), conforme mostrado na figura que segue. O próprio site contém um banco de dados de grids disponíveis para download no formato txt que será utilizado.

Além dos grids é fornecida como entrada uma lista de palavras (arquivo `lista_palavras.txt`), que foram extraídas do site <https://xd.saul.pw/data>, que compilou mais de 6000 palavras cruzadas no The New York Times desde 1965.

Grid:



Arquivo texto:

```
.?.????  
?????.?.  
.?.?.?.  
????????  
.?.?.?.  
.?.????  
?????.?.
```

Saída: a saída do programa deverá ser um arquivo texto contendo o grid fornecido como entrada preenchido com as palavras fornecidas, ou seja, substituindo os pontos de interrogação do arquivo de entrada pelas palavras escolhidas. O programa deve cronometrar o tempo gasto para preencher todo o grid e exibir essa informação em tela ao final do processamento. Além disso, o programa deve conter um registro que permita observar a execução do algoritmo, seja em tela ou em arquivo.

Observações: O grupo é livre para decidir a linguagem de programação a ser utilizada na implementação. Além disso, o grupo pode utilizar bibliotecas que implementem os algoritmos de CSP vistos em aula, desde que o grupo estude a documentação / implementação destas bibliotecas e saiba descrever os algoritmos que estão sendo utilizados no trabalho.

É permitido o uso de ferramentas de IA para gerar código-fonte, desde que: (i) elas sejam usadas somente para tarefas auxiliares e (ii) o grupo indique com comentários no código quais os trechos gerados por IA. O mesmo vale para código que não foi produzido pelo grupo. Se for identificado plágio de qualquer natureza a nota do trabalho será zerada.

Testes

O arquivo testes.zip contém três diagramas diferentes para fins de testes dos algoritmos desenvolvidos. É de responsabilidade do grupo rodar seus algoritmos sobre esses arquivos e entregar os resultados obtidos (arquivo de saída + saídas do algoritmo) para avaliação. Os tempos de execução obtidos pelo grupo em cada teste devem ser incluídos nos slides de apresentação do trabalho.

Apresentação

A apresentação do trabalho será feita para toda a turma nas aulas previstas no cronograma. Cada grupo terá 10 minutos para se apresentar. Deverão ser confeccionados alguns slides descrevendo a forma como o problema foi tratado, quais algoritmos foram utilizados, detalhes relevantes de implementação e testes. Após isso o grupo deverá realizar uma demonstração da solução desenvolvida para a turma.