

Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação – SCC

SCC-218 – Algoritmos Avançados e Aplicações

Professor Gustavo Batista
gbatista@icmc.usp.br

Estagiários PAE
Antonio Parmezan
antoniorafaelparmezan@gmail.com
Felipe Louza
felipelouza@gmail.com

Monitor: Denis Reis
denismr@gmail.com

Projeto – Backtracking
Data de entrega: 07/10

O objetivo deste projeto é implementar um sistema de coloração de mapas utilizando a técnica de força bruta (*backtracking*) e as heurísticas de verificação adiante, mínimos valores remanescentes e grau.

Este projeto possui três partes descritas a seguir:

Parte I – Implementação (50% da nota final)

Esta parte do projeto consiste em implementar um algoritmo força bruta para coloração de grafos planares (mapas) e as heurísticas discutidas em aula. A implementação deve seguir as seguintes diretrizes:

1. A implementação deve ser capaz de trabalhar com qualquer mapa com até 100 regiões;
2. A única restrição existente é que duas regiões vizinhas não podem ter uma mesma cor. As restrições devem ser representadas por meio de um grafo;
3. A implementação deve ser capaz de utilizar *backtracking* simples ou com três composições de heurísticas de poda:
 - a. *Backtracking* simples, sem poda;
 - b. *Backtracking* com verificação adiante. *Backtracking* é realizado quando uma variável fica sem nenhum valor disponível;

- c. *Backtracking* com verificação adiante e mínimos valores remanescentes. Idem ao anterior com a adição da heurística MVR para decidir a próxima variável;
 - d. *Backtracking* com verificação adiante, MVR e grau. Idem ao anterior com o uso da heurística grau nos casos de empate de MVR.
4. A sua implementação deve ser uma única função de *backtracking* com *flags* que permitem ligar/desligar cada uma das heurísticas citadas.

Parte II – Avaliação de Funcionamento (20% da nota final)

Esta avaliação consiste em avaliar o correto funcionamento do seu programa. Para isso o programa deve ler uma entrada e imprimir uma saída em formatos rígidos descritos a seguir.

O programa deve ler uma entrada que consiste em uma linha com um número N ($0 < N < 101$) que representa o número de regiões no mapa a ser colorido. Nas próximas N linhas seguem N listas contendo as regiões vizinhas. Cada lista está em uma linha diferente no formato “**Nome da região: Região vizinha 1, Região vizinha 2, ...**” etc. As regiões vizinhas são separadas por vírgulas e a lista termina com um ponto final. Segue um exemplo:

```
26
Acre: Amazonas.
Amapa: Para.
Amazonas: Acre, Rondonia, Mato Grosso, Para, Roraima.
...
```

A saída do seu programa deve ter a coloração do grafo no seguinte formato: O programa deve imprimir uma lista com N linhas, sendo que cada linha deve conter apenas “Região: cor.”, ou seja, o nome da região e a cor atribuída a ela. Por exemplo:

```
Acre: Azul.
Amapa: Azul.
Amazonas: Vermelho.
...
```

Como é provado que qualquer grafo planar pode ser colorido com quatro cores, utilize nos seu programa as cores “Azul”, “Amarelo”, “Verde” e “Vermelho”.

Parte III – Avaliação de Desempenho (30% da nota final)

A avaliação de desempenho consiste em comparar os tempos de execução do algoritmo de força bruta simples e com o uso de heurísticas de poda para o mapa dos Estados Unidos da América.

Como medir o desempenho em termos de tempo de execução utilizando um relógio pode levar a resultados que variam com a linguagem de programação, computador e outros fatores, neste projeto o desempenho será avaliado pelo número de atribuições de cores realizadas durante toda a busca até se encontrar uma primeira solução.

Como o número de atribuições pode ser muito grande para as podas mais fracas, o seu programa deve abortar a busca quando o número de atribuições de cores exceder 10^6 , mas apenas se for necessário.

Faça um curto relatório (1 a 2 páginas) explicando a sua implementação e os resultados obtidos na Parte III.