

Trabalho Prático 2 - Surpresa na Prova

Introdução

Um professor de Teoria de Grafos do MIT, encucado com a similaridade das respostas de sua última prova, decidiu fazer um estudo inovador em sua turma.

Ele distribuiu uma prova onde havia apenas uma questão de múltipla escolha, com N opções. Cada aluno deveria marcar quais opções ele achasse correta, podendo marcar desde apenas uma opção até todas as N . O objetivo do professor não era corrigir se o aluno respondeu certo ou errado, e sim avaliar qual a semelhança entre as resposta dada por cada aluno.

Ao corrigir as provas o professor então analisou as respostas. Ele mapeou esse estudo em um grafo, para ser possível avaliar tal semelhança. Com isso, o professor quer descobrir qual o maior conjunto de alunos que não marcaram nenhuma resposta igual, com o objetivo de identificar o maior número de alunos que não trocaram respostas entre si.

Soluções

Para este trabalho, você deverá apresentar duas soluções: uma que apresente a solução ótima (solução exata) e outra solução - não necessariamente ótima - obtida através de uma heurística. Caso você consiga desenvolver - e provar - que sua heurística é um algoritmo aproximativo, você ganhará três pontos extras!

Entrada

O arquivo de testes pode conter diversas instâncias. A primeira linha do arquivo contém um inteiro com o número de instâncias presentes no teste. Na primeira linha de cada instância estarão dois inteiros A e N , separados por um espaço em branco, onde A é o número de alunos e N é o número de opções de respostas para a questão da prova. As próximas N linhas conterão os números dos alunos que marcaram cada uma das opções, separados por um espaço em branco. Ou seja, na i -ésima linha estará os números dos alunos que marcaram aquela i -ésima opção. Cada aluno será identificado por um número que vai de 1 ao valor de A .

A seguir temos um exemplo de entrada do programa:

Entrada:

```
2
5 8
1 4 5
2 5
3 4
```

1 4
3 4 5
3 5
1 5
1 4
6 10
1 2 4
1 6
2 6
1 4
1 2 5
2 4
1 3
1 2 5
3
5

Nesse exemplo, temos que na primeira instância 5 alunos e a prova apresentava 8 opções de escolha. Os alunos 1, 4 e 5 marcaram a opção 1. Os alunos 2 e 5 marcaram a opção 2 e assim por diante. Já na segunda instância temos 6 alunos e 10 opções de resposta, onde os alunos 1, 2 e 4 marcaram a opção 1 e assim por diante. Além disso, o professor observou que na prova nenhuma opção deixou de ser marcada, ou seja, sempre haverá pelo menos um aluno que marcou uma determinada opção.

Saída

Para cada instância de teste, deverá ser impresso no arquivo de saída a quantidade de alunos que fazem parte do maior conjunto que não apresentou nenhuma resposta igual. A saída de cada instância deve estar em uma linha, sem espaços em branco no final. Entre duas instâncias distintas NÃO deve existir espaço em branco.

Veja a saída do exemplo anterior:

Saída:

3
4

Entrega

- A data de entrega desse trabalho é **04/04/2014**.
- A penalização por atraso obedece à seguinte fórmula $2^{d-1}/0.32\%$, onde d são os dias úteis de atraso.
- Submeta apenas um arquivo chamado `<numero_matricula>_<nome>.zip`. Não utilize espaços no nome do arquivo. Ao invés disso utilize o caractere `'_'`.
- Não inclua arquivos compilados ou gerados por IDEs. **Apenas** os arquivos abaixo devem estar presentes no arquivo zip.
 - Makefile
 - Arquivos fonte (*.c e *.h)
 - Documentacao.pdf
- Não inclua **nenhuma pasta**. Coloque todos os arquivos na raiz do zip.
- Siga rigorosamente o formato do arquivo de saída descrito na especificação. Tome cuidado com whitespaces e formatação dos dados de saída
- **NÃO SERÁ NECESSÁRIO ENTREGAR DOCUMENTAÇÃO IMPRESSA!**
- Será adotada **média harmônica** entre as notas da **documentação e da execução**, o que implica que a nota final será 0 se uma das partes não for apresentada.

Documentação

A documentação não deve exceder 10 páginas e deve conter pelo menos os seguintes itens:

- Uma **introdução** do problema em questão.
- **Modelagem e solução proposta** para o problema. O algoritmo deve ser explicado de forma clara, possivelmente através de pseudo-código e esquemas ilustrativos.
- **Análise de complexidade** de tempo e espaço da solução implementada.
- **Experimentos** variando-se o tamanho da entrada e quaisquer outros parâmetros que afetem significativamente a execução.
 - Espera-se a utilização de tabelas e gráficos, com suas respectivas análises.
- Especificação da(s) **máquina(s) utilizada(s)** nos experimentos realizados.
- Uma breve **conclusão** do trabalho implementado.

Código

- O código deve ser obrigatoriamente escrito na **linguagem C**. Ele deve compilar e executar corretamente nas máquinas Linux dos laboratórios de graduação.
- O utilitário ***make*** deve ser utilizado para auxiliar a compilação, um arquivo *Makefile* deve gerar dois executáveis com o nome tp2e e tp2h que deverão obrigatoriamente ser capazes de receber o nome de um arquivo de entrada - de onde serão lidas as instâncias do problema - e o nome de um arquivo de saída - onde serão gravadas as soluções. O comando para executar as soluções deverá seguir o exemplo abaixo:

```
./tp2e input.txt output.txt
```

```
./tp2h input.txt output.txt
```

- As estruturas de dados devem ser **alocadas dinamicamente** e o código deve ser **modularizado** (divisão em múltiplos arquivos fonte e uso de arquivos cabeçalho .h)
- **Variáveis globais** devem ser evitadas.
- Parte da correção poderá ser feita de forma automatizada, portanto **siga rigorosamente os padrões de saída especificados**, caso contrário sua nota pode ser prejudicada.
- **Legibilidade e boas práticas** de programação serão avaliadas.