

Trabalho Prático 0

Este trabalho prático tem por objetivo exercitar primitivas básicas da Linguagem C e iniciar a discussão sobre problemas complexos e suas soluções.

1 Definição do Problema

Este trabalho aborda o problema clássico do Casamento Estável (*Stable Marriage Problem*, SMP) que consiste em encontrar um *matching* estável (*stable matching*) entre dois conjuntos distintos de elementos. Os conjuntos têm o mesmo tamanho e cada elemento possui uma lista de preferências sobre o outro conjunto.

O SMP é usualmente formulado como dois conjuntos de homens e mulheres com n elementos cada, sendo que cada indivíduo possui uma lista ordenada de preferências (também de tamanho n) sobre os indivíduos do sexo oposto. Dado um indivíduo i , a pessoa que ocupa a primeira posição de sua lista de preferências representa o pretendente o qual i prefere se casar. De forma análoga, o último elemento dessa lista indica o pretendente mais rejeitado por i .

Um casamento é considerado estável quando **NÃO** há duas pessoas de sexo oposto, em que **ambas** desejariam estar casadas uma com a outra, ao invés de seus atuais parceiros. Todos os casamentos são considerados estáveis, se não há indivíduos nessa situação. A Figura 1 exemplifica um cenário de casamentos não estáveis. Se **ambas** as situações ilustradas nas Figuras 1(a) e 1(b) ocorrerem, os casamentos entre AC e BD são considerados instáveis.

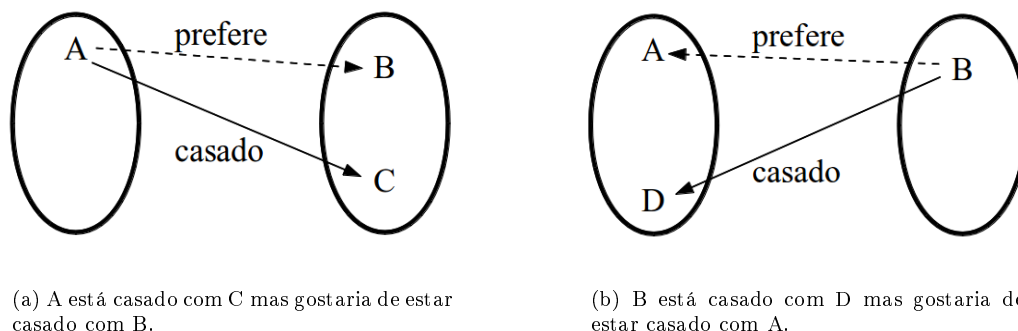


Figura 1: Exemplo de casamentos **NÃO** estáveis.

O aluno deve elaborar um algoritmo que represente uma heurística capaz de encontrar uma solução válida para o SMP, dada uma instância fornecida como entrada do problema. A implementação do algoritmo **DEVE** utilizar listas encadeadas com alocação de memória dinâmica.

1.1 Entrada e Saída

O arquivo executável deve ser chamado de *tp0* e deve receber como entrada o arquivo *input.txt* com a lista de preferências de todos os indivíduos.

O arquivo *input.txt* deve conter as seguintes informações, na ordem indicada abaixo:

1. número de instâncias do problema;
2. tamanho n dos conjuntos considerados;
3. n linhas com as preferências de cada homem;
4. n linhas com as preferências de cada mulher.

Cada indivíduo é representado por um identificador único (*id*), por sexo. Cada *id* é um valor inteiro entre 1 e *n*. Sendo assim, cada linha possui todos os *ids* do sexo oposto ordenados pela preferência do usuário correspondente.

Exemplo de entrada para 8 indivíduos (4 homens - *ids* de 1 a 4, e 4 mulheres - *ids* de 1 a 4):

```
1 // número de instâncias.
4 // número n de indivíduos em cada conjunto.
2 4 1 3 // homem de id = 1 prefere mulher de id = 2, seguido da mulher de id = 4, e assim por diante.
4 1 2 3
2 3 4 1
2 3 4 1
2 3 4 1 // mulher de id = 1 prefere homem de id = 2, seguido do homem de id = 3, e assim por diante.
3 4 2 1
3 2 1 4
3 2 4 1
```

A saída do programa deve ser impressa em um arquivo *output.txt* e deve conter as seguintes informações, na ordem indicada abaixo:

1. indicação dos casamentos finais de todos os indivíduos;
2. valor da satisfabilidade* geral dos indivíduos;
3. valor da satisfabilidade masculina;
4. valor da satisfabilidade feminina;

***Satisfabilidade** é uma métrica expressa como a média das posições que os maridos e/ou esposas ocupam na lista de preferências de seus respectivos parceiros. Isto é, se uma mulher se casa com seu pretendente preferido, ela contribui em 1 para a satisfabilidade. No entanto, caso ela se case com a terceira opção de sua lista de preferências, sua contribuição será de 3. Sendo assim, quanto menor o valor dessa métrica, maior a satisfação dos indivíduos considerados.

Exemplo de saída (correspondente ao exemplo de entrada descrito anteriormente):

```
2 4 // homem de id = 2 se casou com mulher de id = 4.
1 1
3 2
4 3
2,25 // satisfabilidade geral.
1,75 // satisfabilidade masculina.
2,75 // satisfabilidade feminina.
```

Os comentários nos exemplos de entrada e saída foram apresentados apenas por propósitos didáticos, logo eles **não** devem ser incluídos nos arquivos para submissão. Além disso, entrada e saída padrão devem seguir rigorosamente o formato descrito, inclusive no caso de múltiplas instâncias. Sendo assim, em ambos os arquivos, dados de diferentes instâncias devem ser apresentados na mesma ordem e em sequência (na linha seguinte do último dado da instância anterior). Instâncias distintas para o SMP devem ser geradas pelo próprio aluno para testar e avaliar seu algoritmo.

2 O que deve ser entregue:

2.1 Documentação: deve abranger pelo menos os seguintes pontos

- Introdução do problema apresentado.
- Modelagem e solução do problema: explique o algoritmo utilizado para resolver o problema e a sua complexidade de tempo e espaço.
- Principais decisões de implementação.
- A documentação não pode exceder 10 páginas.

2.2 Código:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios de graduação do DCC;
- Deve ser **obrigatoriamente** escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras **não** serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado (ou seja, dividido em múltiplos arquivos fonte e fazendo uso de arquivos cabeçalho - .h);
- O utilitário Make deve ser utilizado para compilar o programa (**o arquivo de makefile deve ser submetido juntamente com o código fonte**);
- A saída deve ser impressa seguindo estritamente o formato da especificação, caso contrário o resultado será considerado errado;
- O arquivo executável deve ser chamado de **tp0**. Não serão aceitos outros nomes de executáveis além do mencionado;
- Faça seu código de forma legível.

2.3 Entrega:

- Data de entrega: 22/03/2012
- Submissão: a documentação e o código do trabalho devem ser submetidos ao minha.ufmg.br. Para isso, compacte os dois (formato tar.gz ou zip) e faça a submissão. Teste seu arquivo compactado antes de enviá-lo.
- Apenas a documentação também deve ser entregue impressa na secretaria do DCC. Não coloque nos escaninhos dos professores. A documentação deve ser entregue para a secretária e então colocada no envelope de AEDS3. **Trabalhos que não tiverem a documentação entregue na secretaria, dentro do prazo de entrega, receberão nota 0.**
- Será postada uma planilha no Moodle sobre a entrevista do trabalho, leia-a e siga as orientações para o agendamento da sua entrevista.
- Será adotado média harmônica entre a pontuação obtida na execução e na documentação do TP, o que implica em valor zero caso alguma das partes não seja apresentada.

Referências

Projeto de Algoritmos com implementação em Pascal e C. Nívio Ziviani - <http://www.dcc.ufmg.br/algoritmos/>.

Introduction to Algorithms. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. The MIT Press, 3rd edition, 2009.

Stable Marriage Problem - http://en.wikipedia.org/wiki/Stable_marriage_problem.

Site interativo do SMP - <http://mathsite.math.berkeley.edu/smp/smp.html>.