Casamentos Estáveis - Algoritmos em Python

Luciano Monteiro de Castro, Renato Madeira

17 de Junho de 2013

Conteúdo

1 Resumo

Este documento contém instruções para utilização dos algoritmos contidos no arquivo `casamentosestaveis.py`,
encapsulados no formato de funções. O arquivo contém descrições das funções em si, de forma que nosso enfoque aqui será o de adotar o ponto de vista dos possíveis usuários destes algoritmos, de acordo com a natureza de seu interesse no problema dos casamentos estáveis.

2 O Problema dos Casamentos Estáveis

Consiste em tentar casar de maneira satisfatória as pessoas de um grupo de n homens e n mulheres, onde cada uma das pessoas deve classificar todas as do sexo oposto por ordem de preferência para o casamento. Neste caso, um emparelhamento será dito instável se houver um homem e uma mulher que prefiram um ao outro a seus atuais parceiros. É claro que homens e mulheres podem ser substituídos por outros tipos de grupo, e a solução deste problema aplicada a diversas outras situações em que se busque emparelhamentos satisfatórios. Veja mais em
`http://pt.wikipedia.org/wiki/Problema_do_emparelhamento_estável`

3 Determinação de um emparelhamento (casamento) estável

Se você deseja utilizar este pacote para resolver uma instância concreta do problema dos casamentos estáveis, mesmo que não tenha qualquer experiência em Python, pode fazê-lo através das funções `casar` ou `casarp`, dependendo do formato em que preferir obter a solução. Para isso, siga os seguintes passos:

3.1 Crie listas numeradas de preferências para ambos os grupos

Para facilitar, manteremos a convenção usual e chamaremos os membros de um grupo de "homens"e os do outro grupo de "mulheres". Você deve numerar tanto os homens como as mulheres utilizando os números naturais de 0 a $n-1$, onde n representa a quantidade de pessoas em cada grupo. Depois, cada homem deve listar as mulheres em sua ordem de preferência, começando por sua preferida, e cada mulher deve fazer o mesmo para os homens. As mulheres e os homens nas listas devem ser representados pelos seus números. Dada a natureza do algoritmo usado na solução, o emparelhamento encontrado será o melhor possível para os homens, então você pode querer escolher como grupo dos homens o que mais lhe convier.

Exemplo:

Homens: 0. André, 1. Bruno, 2. Carlos.

Mulheres: 0. Ana, 1. Bianca, 2. Claudia.

Se a ordem de preferências de André é Bianca, Claudia, Ana, então a sua lista será 1, 2, 0. Se a lista de Ana é 2, 1, 0, sua ordem de preferências é Carlos, Bruno, André.

3.2 Crie um arquivo de texto que traduza as listas de preferências

Esta etapa requer absoluta precisão. O arquivo de texto deve conter as $2n$ listas de preferências, começando pelos homens e depois pelas mulheres. Os números em uma mesma lista devem ser separados por espaços, e uma lista deve ser separada da outra por ponto e vírgula (;). O arquivo deve constituir-se de uma única linha de texto.

Exemplo: Com os homens e mulheres do exemplo anterior, a única linha de nosso arquivo de texto deve começar com a lista de André: `1 2 0`. Deve seguir-se um ponto e vírgula e a lista de Bruno, e assim por diante. Ao final o conteúdo do arquivo será algo assim:

```
1 2 0; 0 1 2; 1 0 2; 2 1 0; 0 1 2; 0 2 1
```

Observe que as 6 listas de preferências são facilmente identificáveis, e estão na ordem André, Bruno, Carlos, Ana, Bianca, Claudia. Há um arquivo como este no repositório com o nome `exemplo.cas`.

3.3 Inicie uma sessão de Python

Para isso você precisa instalar uma versão do Python em seu computador. Ver, por exemplo, `www.python.org`.

3.4 Execute a função CASARP

Esta função imprime diretamente na tela a lista dos pares de um emparelhamento estável, cada par contendo primeiro o número do homem, e depois o número da mulher.

Para executá-la, digite primeiro

```
>>> import casamentosestaveis as cas
```

seguido da tecla 'enter'.

(você precisa baixar o arquivo `casamentosestaveis.py` contido neste repositório, e gravá-lo em uma pasta acessível ao Python, por exemplo raiz ou home).
Depois execute a função `casarp` sobre o arquivo contendo as listas de preferências. Para isso digite
`>>> cas.casarp('nomedoarquivo'`
seguido de 'enter'. Substitua "nomedoarquivo"pelo nome correto do arquivo, mas não esqueça de fazê-lo entre ' e '.
Exemplo: Continuando com o mesmo exemplo dos itens anteriores, eis o que executamos e obtemos:

```
>>> import casamentosestaveis as cas
import casamentosestaveis as cas
>>> cas.casarp('exemplo.cas')
cas.casarp('exemplo.cas')
[0, 1]
[1, 2]
[2, 0]
>>>
```

Ou seja, o homem 0, André, fica emparelhado com a mulher 1, Bianca. Bruno fica com Claudia e Carlos, com Ana.

3.5 Ou execute a função CASAR

A função `casar` retorna uma lista que representa os pares de cada homem, em ordem. É especialmente útil se você pretende usar o resultado do algoritmo para novos processamentos, como por exemplo testes (ver mais adiante).
Exemplo: Usando o mesmo arquivo do item anterior:

```
>>> import casamentosestaveis as cas
import casamentosestaveis as cas
>>> cas.casar('exemplo.cas')
cas.casar('exemplo.cas')
[1, 2, 0]
>>>
```

Esta lista significa que os pares dos homens $0, 1, 2$ são respectivamente as mulheres $1, 2, 0$, que é a mesma solução obtida anteriormente.

4 Realização de Testes

As funções definidas em `casamentosestaveis.py` permitem a realização de testes de correção e performance dos algoritmos implementados. A seguir oferecemos alguns exemplos. Consultando os comentários sobre cada função inseridos no próprio arquivo podem ser desenvolvidos outros testes.

4.1 Teste da estabilidade de um emparelhamento

Este provavelmente é o teste mais importante para as implementações que desenvolvemos, pois significa responder à pergunta: "O código funciona?".

Há uma função específica para isso, nomeada `estabilidade`. Ela recebe uma lista `h`, representando os pares de cada homem em ordem, e uma matriz de preferências `pref`, que se supõe já estar no formato convertido (ver mais detalhes nos comentários do próprio arquivo).
O funcionamento deste teste é muito simples: para cada homem `i`, percorre-se sua lista de preferências até chegar ao seu atual par, ou seja, verificam-se todas as mulheres que ele prefere ao seu par. Se alguma delas também prefere o homem `i` ao seu atual par, uma instabilidade é detectada. Caso contrário, o emparelhamento é determinado estável.
Se você está desenvolvendo ou desenvolveu sua própria implementação de solução para este problema, pode usar esta função para testar a correção do seu trabalho.
Exemplo: Continuando com nosso arquivo `exemplo.cas`, vamos testar se o casamento obtido é realmente estável. Para isso, vamos guardar a solução em uma lista `h`, e vamos ler e converter o arquivo para guardar a informação na matriz convertida `pref`.

```
>>> import casamentosestaveis as cas
import casamentosestaveis as cas
>>> h = cas.casar('exemplo.cas')
h = cas.casar('exemplo.cas')
>>> h
h
[1, 2, 0]
>>> pref = cas.converter(cas.ler('exemplo.cas'))
pref = cas.converter(cas.ler('exemplo.cas'))
>>> pref
pref
array([[1, 2, 0],
       [0, 1, 2],
       [1, 0, 2],
       [2, 1, 0],
       [0, 1, 2],
       [0, 2, 1]])
>>> cas.estabilidade(h,pref)
cas.estabilidade(h,pref)
casamento estável!
>>>
```

Recebemos a mensagem de que o casamento é estável.
O que ocorre se o casamento não for estável? Vamos modificar `h`, trocando as mulheres 1 e 2 de posição:

```
>>> h[0] = 2
h[0] = 2
>>> h[1] = 1
h[1] = 1
>>> h
h
[2, 1, 0]
>>> cas.estabilidade(h,pref)
cas.estabilidade(h,pref)
```

```
['instabilidade', 0, 1]
>>>
```

O programa indica a primeira instabilidade encontrada, a "instabilidade $(0, 1)$", que significa que o homem 0 e a mulher 1 prefeririam estar juntos.

4.2 Teste do algoritmo com matrizes aleatórias

A função `aleatoria` recebe um inteiro positivo n e gera uma matriz de preferências aleatória para n homens e n mulheres, ou seja, uma matriz $2n \times n$. Esta matriz pode ser suposta no formato padrão ou já no formato convertido.
A função é muito útil como teste, já que é tedioso criar listas de preferências para n muito grande. A forma mais básica de testar o algoritmo com esta função é criar uma matriz, executar o algoritmo principal sobre ela (função `casamentos`) e depois testar a estabilidade do resultado:

```
>>> import casamentosestaveis as cas
import casamentosestaveis as cas
>>> pref = cas.aleatoria(15)
pref = cas.aleatoria(15)
>>> pref
pref
matrix([[ 2,  7, 10,  0, 12,  1,  4,  5,  8, 14, 13, 11,  6,  3,  9],
        [13, 11, 10, 12,  0,  2,  4,  1,  8,  9,  3,  7, 14,  6,  5],
        [ 8,  4,  2, 12,  3, 10, 11, 13,  9,  1,  7,  5,  0, 14,  6],
        [ 2,  5,  7, 10,  3, 14,  4,  6, 12, 13,  8, 11,  0,  9,  1],
        [ 9, 11,  6,  3, 14,  8, 13,  0,  4, 12,  5,  1,  2, 10,  7],
        [ 3,  0, 14, 10,  9,  6, 12,  5,  4,  2,  8, 13,  1, 11,  7],
        [ 1, 14, 11,  4, 13,  8,  5,  2, 12,  9,  3, 10,  7,  0,  6],
        [ 1, 14,  6,  7, 13,  2,  3,  9, 12, 11,  4, 10,  8,  0,  5],
        [ 2,  3,  0,  4,  9,  1, 14, 13,  8,  7, 12,  6, 11,  5, 10],
        [ 0,  3,  9,  7,  5,  6,  2,  4, 11, 10,  8,  1, 14, 12, 13],
        [ 6,  0,  2,  9,  8, 10,  5, 11, 13, 14,  1,  7, 12,  4,  3],
        [ 1,  6,  9,  4,  0,  8, 12, 10,  7,  5, 14, 11,  3, 13,  2],
        [12,  3,  6, 13,  0,  1,  2,  8, 11,  5,  4,  9, 14,  7, 10],
        [11,  0, 14, 13, 10,  8,  4, 12,  2,  1,  9,  3,  5,  7,  6],
        [ 2,  7, 10,  4,  8,  9,  5, 11,  1,  6,  0, 12,  3, 13, 14],
        [14,  8,  6,  2, 10, 12, 13,  4,  0, 11,  1,  9,  5,  7,  3],
        [14, 11,  9, 13,  4, 10,  1,  3,  7,  6,  8,  0,  2,  5, 12],
        [ 5,  6, 11,  8, 13, 14,  2, 12,  0,  4,  3, 10,  7,  9,  1],
        [ 8,  6,  3, 12,  5,  0,  2,  7, 13, 10,  4, 11,  1,  9, 14],
        [ 3,  4, 11,  2,  7,  6, 13,  9, 14, 12,  5, 10,  0,  1,  8],
        [13,  2, 11,  8,  1,  5,  6,  0, 10,  4,  3, 12,  7, 14,  9],
        [ 1,  7, 10,  4, 11, 14,  2,  9,  3,  6, 12,  0, 13,  8,  5],
        [10,  9,  3,  2,  1,  8, 14, 12,  5, 13,  4,  0,  6, 11,  7],
        [ 6,  2, 12, 14,  8,  3, 10,  0,  7,  5,  4,  9, 11,  1, 13],
        [ 7,  4,  5,  9,  6,  8,  2,  1, 11, 14,  3,  0, 13, 12, 10],
        [14,  2,  1,  3,  0, 11,  5,  8,  7,  9, 10,  4, 12,  6, 13],
        [ 1, 12, 10,  8,  3,  4,  6,  5,  9,  0, 13, 11,  2,  7, 14],
```

```
        [ 9,  5, 13, 14,  1, 10,  0, 12,  2,  8,  4,  6,  3,  7, 11],
        [ 8, 10, 14,  3, 11,  2,  4,  7,  6,  9,  0,  1, 12,  5, 13],
        [ 0, 14,  2,  6,  5,  3, 10, 11,  7, 12,  4, 13,  8,  9,  1]])
>>> h = cas.casamentos(pref)
h = cas.casamentos(pref)
>>> h
h
[4, 13, 8, 7, 9, 3, 14, 6, 2, 5, 0, 1, 12, 11, 10]
>>> cas.estabilidade(h,pref)
cas.estabilidade(h,pref)
casamento estável!
>>>
```

Podemos automatizar um grande número de testes como este, com `n` cada vez maior, até nos sentirmos confortáveis quanto à correção do nosso trabalho.

5 Próximos Passos

Gostaríamos de implementar um aplicativo base web que permitisse a qualquer pessoa com acesso a um navegador inserir listas de preferências e obter um emparelhamento estável. Isto poderia ser útil, por exemplo, para os tradicionais bailes de debutantes em que a aniversariante convida 14 amigos e 14 amigas para dançar a valsa.
Também gostaríamos de avançar na elaboração de uma sugestão eficaz para o Sisu brasileiro.