

MC102 - Algoritmos e Programação de Computadores

Lab 12

Data da Primeira Chance: 26 de junho de 2023

Peso: 3

Duvido, ou Mentira, é um jogo de baralho no qual seu objetivo é descartar todas as cartas da mão para uma pilha. Ele se inicia dividindo as 52 cartas de um baralho entre os jogadores participantes. No início do jogo, o jogador inicial escolhe o maior número de cartas do mesmo número/letra para descartar. Ele desce essas cartas viradas com face para baixo (escondidas) na pilha e diz em voz alta qual carta é. Nesse momento o jogador pode blefar, ou seja, dizer em voz alta uma carta diferente da que está jogando de fato. Os demais jogadores da partida podem acreditar ou duvidar dele, dizendo “Duvido!”

Caso todos acreditem, as cartas do jogador permanecem na pilha e é a vez do próximo jogador realizar os mesmos passos para descartar cartas de sua mão. Caso alguém duvide, existem dois cenários possíveis:

- 1) O jogador realmente jogou as cartas que disse: nesse caso o jogador que duvidou deve pegar todas as cartas da pilha e colocá-las na sua mão.
- 2) O jogador mentiu sobre as cartas que jogou: nesse caso o jogador mentiroso fica com todas as cartas da pilha na sua mão.

As cartas possuem a seguinte ordem de força: **A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K** e teremos a indicação dos naipes para representar todas as 52 cartas do baralho. Os naipes serão representados pela primeira letra sendo **O** = ouros, **E** = espadas, **C** = copas e **P** = paus, sendo essa a ordem de força crescente dos naipes. Assim, as cartas na entrada terão o formato 6O, 6E, 6C e 6P.

O jogo progride por rodadas, onde os jogadores colocam cartas com as faces viradas para baixo na pilha de descarte. Em cada rodada, as cartas jogadas devem seguir a ordem crescente de força a partir da carta jogada (inclusive ela). Isso significa que se o jogador jogar um J, o próximo jogador terá como opções de jogada as cartas J, Q ou K e, caso não possua nenhuma delas, terá de blefar. Como as jogadas são realizadas sem especificação dos naipes e com a face virada para baixo, não é necessário considerar a força dos naipes durante uma jogada. Porém, essa informação deve ser considerada para ordenação da mão dos jogadores.

Ao início de cada rodada teremos o começo de uma nova sequência de força de cartas a ser seguida a partir do primeiro descarte. Uma rodada é finalizada quando a pilha for zerada após um “Duvido!”. A pilha de descarte é mantida até o final do jogo, isto é, todas as cartas descartadas desde a primeira rodada se acumulam e a pilha só fica sem cartas se estas forem adicionadas à mão de alguém.

Sumarizando a lógica do jogo, para esse laboratório, temos os seguintes passos:

1. Temos como entrada a mão de todos os jogadores. Os jogadores fazem suas jogadas na ordem da entrada.
2. O primeiro jogador realiza sua jogada colocando na pilha de descarte todas as cartas de mesmo número/letra de menor força e dizendo qual carta é.
3. Os outros jogadores podem ou não duvidar que as cartas são do tipo anunciado. Caso ocorra um “Duvido!” e se **todas** as cartas jogadas contiverem o número/letra anunciado, o acusador leva todas as cartas da pilha, inclusive o descarte atual. Se **qualquer** das cartas jogadas não for do tipo anunciado, o jogador que fez o descarte leva todas as cartas da pilha, inclusive as cartas que tentou descartar. Após um “Duvido!” o próximo jogador na ordem da entrada faz sua jogada, iniciando uma nova rodada.
4. Após o primeiro descarte, os próximos jogadores devem sempre realizar jogadas que respeitem a ordem crescente de força das cartas, incluindo a(s) carta(s) recém descartada(s). Após cada descarte, o passo 3 é efetuado.
5. O jogo termina quando algum jogador consegue descartar todas as cartas de sua mão, sendo declarado o vencedor.

Entrada

A entrada do programa é constituída pelas seguintes informações, em ordem:

1. Um inteiro **J**, representando a quantidade de jogadores presentes.
2. **J** linhas contendo a mão inicial de cada jogador.
3. Um inteiro **N** representando quantas jogadas acontecem antes de acontecer um “Duvido”. Ou seja, a cada N jogadas feitas, o jogador imediatamente após o que efetuou a última jogada deve confrontá-lo.

Saída

Todo início de rodada deve-se imprimir as **mãos dos jogadores em ordem decrescente** de força das cartas (ordenadas) e a pilha de descarte na ordem em que foi construída (desordenada) ou vazia. Deve-se considerar também a força dos naipes para a ordenação da mão dos jogadores. Note que no início do jogo, a pilha de descarte sempre estará vazia.

Exemplo:

```
Jogador 1
Mão: 5C 4E 3P 3C 3O
Jogador 2
Mão: KO 6E 2C AP AO
Jogador 3
Mão: 7C 5E 5O 2O
Pilha: 2P AC KP JE JC 9P 9O 8C 8P
```

A cada jogada realizada, deve-se imprimir qual jogador está fazendo a jogada, qual carta ele diz jogar e a pilha atual de descarte atualizada com as cartas que foram recentemente adicionadas. **Caso algum jogador duvide, deve-se também imprimir a mão de todos os jogadores e a pilha de descarte após essa ação.** Exemplo:

```
[Jogador 1] 3 carta(s) 3
Pilha: 2P AC KP JE JC 9P 9O 8C 8P 3O 3C 3P
Jogador 2 duvidou.
Jogador 1
Mão: KP QE JC JE 9P 9O 8P 8C 7E 5C 4E 3P 3C 3E 2P AC
Jogador 2
Mão: KO 6E
Jogador 3
Mão: 7C 5E 5O 2O
Pilha:
```

Se ao final de alguma jogada o jogador ficar com a mão vazia, ele será o vencedor do jogo e deve-se imprimir a mensagem "Jogador **X** é o vencedor."

Observações:

- **O jogador deve blefar sempre que não tenha nenhuma carta que respeite a ordem de força que deve jogar.** O blefe será realizado jogando todas as cartas de mesmo número/letra mais fracas da mão do jogador e dizendo ser a carta do número/letra necessária.
- Deve-se **manter sempre as mãos dos jogadores ordenadas de forma decrescente** de força das cartas, considerando a força dos naipes.
- Lembre-se de que as jogadas devem ser feitas seguindo a ordem de força, então após a jogada anterior, deve-se buscar a próxima carta mais forte que ela na mão do próximo jogador para fazer a jogada seguinte. Caso não exista, blefe.
- Os métodos de ordenação e busca devem ser implementados, isto é, não serão permitidos os usos de funções prontas. Para busca deve-se obrigatoriamente utilizar **Busca Binária**. Pode-se utilizar qualquer algoritmo de ordenação.

Exemplos

Exemplo 1:

Entrada

```
3
QE, 2C, KO
AO, 2E, 2P
AP, AE, AC
3
```

Saída

```
Jogador 1
Mão: KO QE 2C
Jogador 2
Mão: 2P 2E AO
Jogador 3
Mão: AP AC AE
Pilha:
[Jogador 1] 1 carta(s) 2
Pilha: 2C
[Jogador 2] 2 carta(s) 2
```

```
Pilha: 2C 2E 2P
[Jogador 3] 3 carta(s) 2
Pilha: 2C 2E 2P AE AC AP
Jogador 1 duvidou.
Jogador 1
Mão: KO QE
Jogador 2
Mão: AO
Jogador 3
Mão: 2P 2C 2E AP AC AE
Pilha:
[Jogador 1] 1 carta(s) Q
Pilha: QE
[Jogador 2] 1 carta(s) Q
Pilha: QE AO
Jogador 2 é o vencedor!
```

Exemplo 2:

Entrada

```
4
QE, QC, 10P, 2P, 7E, AC
6C, 5C, 3E, 6P, 3O, 8E
6E, KP, 4C, QP, 2C, 10E
9O, KE, 3C, JO, AO, 4O
10
```

Saída

```
Jogador 1
Mão: QC QE 10P 7E 2P AC
Jogador 2
Mão: 8E 6P 6C 5C 3E 3O
Jogador 3
Mão: KP QP 10E 6E 4C 2C
Jogador 4
Mão: KE JO 9O 4O 3C AO
Pilha:
[Jogador 1] 1 carta(s) A
Pilha: AC
[Jogador 2] 2 carta(s) 3
Pilha: AC 3O 3E
[Jogador 3] 1 carta(s) 4
Pilha: AC 3O 3E 4C
[Jogador 4] 1 carta(s) 4
Pilha: AC 3O 3E 4C 4O
[Jogador 1] 1 carta(s) 7
Pilha: AC 3O 3E 4C 4O 7E
[Jogador 2] 1 carta(s) 8
Pilha: AC 3O 3E 4C 4O 7E 8E
[Jogador 3] 1 carta(s) 10
```

```

Pilha: AC 3O 3E 4C 4O 7E 8E 10E
[Jogador 4] 1 carta(s) J
Pilha: AC 3O 3E 4C 4O 7E 8E 10E JO
[Jogador 1] 2 carta(s) Q
Pilha: AC 3O 3E 4C 4O 7E 8E 10E JO QE QC
[Jogador 2] 1 carta(s) Q
Pilha: AC 3O 3E 4C 4O 7E 8E 10E JO QE QC 5C
Jogador 3 duvidou.
Jogador 1
Mão: 10P 2P
Jogador 2
Mão: QC QE JO 10E 8E 7E 6P 6C 5C 4C 4O 3E 3O AC
Jogador 3
Mão: KP QP 6E 2C
Jogador 4
Mão: KE 9O 3C AO
Pilha:
[Jogador 3] 1 carta(s) 2
Pilha: 2C
[Jogador 4] 1 carta(s) 3
Pilha: 2C 3C
[Jogador 1] 1 carta(s) 10
Pilha: 2C 3C 10P
[Jogador 2] 1 carta(s) 10
Pilha: 2C 3C 10P 10E
[Jogador 3] 1 carta(s) Q
Pilha: 2C 3C 10P 10E QP
[Jogador 4] 1 carta(s) K
Pilha: 2C 3C 10P 10E QP KE
[Jogador 1] 1 carta(s) K
Pilha: 2C 3C 10P 10E QP KE 2P
Jogador 1 é o vencedor!

```

Exemplo 3:

Entrada

```

4
10C, 4E, QP, 3E, KE, AO
KP, JP, 4P, 10P, 4C
4O, JC, QO, 3O, AP, 10O, QE
3C, 10E, JE, AE, QC, JO, KO
7

```

Saída

```

Jogador 1
Mão: KE QP 10C 4E 3E AO
Jogador 2
Mão: KP JP 10P 4P 4C
Jogador 3
Mão: QE QO JC 10O 4O 3O AP

```

Jogador 4
Mão: KO QC JE JO 10E 3C AE
Pilha:
[Jogador 1] 1 carta(s) A
Pilha: AO
[Jogador 2] 2 carta(s) 4
Pilha: AO 4C 4P
[Jogador 3] 1 carta(s) 4
Pilha: AO 4C 4P 4O
[Jogador 4] 1 carta(s) 10
Pilha: AO 4C 4P 4O 10E
[Jogador 1] 1 carta(s) 10
Pilha: AO 4C 4P 4O 10E 10C
[Jogador 2] 1 carta(s) 10
Pilha: AO 4C 4P 4O 10E 10C 10P
[Jogador 3] 1 carta(s) 10
Pilha: AO 4C 4P 4O 10E 10C 10P 10O
Jogador 4 duvidou.
Jogador 1
Mão: KE QP 4E 3E
Jogador 2
Mão: KP JP
Jogador 3
Mão: QE QO JC 3O AP
Jogador 4
Mão: KO QC JE JO 10P 10C 10E 10O 4P 4C 4O 3C AE AO
Pilha:
[Jogador 4] 2 carta(s) A
Pilha: AO AE
[Jogador 1] 1 carta(s) 3
Pilha: AO AE 3E
[Jogador 2] 1 carta(s) J
Pilha: AO AE 3E JP
[Jogador 3] 1 carta(s) J
Pilha: AO AE 3E JP JC
[Jogador 4] 2 carta(s) J
Pilha: AO AE 3E JP JC JO JE
[Jogador 1] 1 carta(s) Q
Pilha: AO AE 3E JP JC JO JE QP
[Jogador 2] 1 carta(s) K
Pilha: AO AE 3E JP JC JO JE QP KP
Jogador 3 duvidou.
Jogador 1
Mão: KE 4E
Jogador 2
Mão:
Jogador 3
Mão: KP QP QE QO JP JC JE JO 3E 3O AP AE AO
Jogador 4
Mão: KO QC 10P 10C 10E 10O 4P 4C 4O 3C
Pilha:
Jogador 2 é o vencedor!

Regras e Avaliação

Nesse laboratório, você não pode usar bibliotecas (isto é, o comando *import*), exceto pela biblioteca *typing* para melhorar a clareza e escrita do código. Também está proibida a utilização das funções: `sort()`, `sorted()`, `index()` e `find()`. Além dessas funções, formas de buscar elementos em listas que não sejam através da Busca Binária não serão aceitas, incluindo-se, mas não se limitando ao uso de comando “in” com essa finalidade. Note que iterar por uma lista utilizando o comando “in” como em “for x in list:” pode ser utilizado.

Seu código será avaliado não apenas pelos testes do CodePost, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código analisaremos nesse laboratório: as implementações dos algoritmos de busca binária e ordenação, o uso apropriado de funções, e de documentação; a escolha de bons nomes de funções e variáveis; a ausência de diversos trechos de código repetidos desnecessariamente. Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

Os casos de testes estão disponíveis através do [link](#).

Submissão

Você deverá submeter no CodePost, na tarefa Lab 12, um arquivo com o nome `lab12.py`. Você pode enviar arquivos adicionais caso deseje para serem incluídos por `lab12.py`. Após a correção da primeira entrega, será aberta uma tarefa Lab 12 - Segunda Chance, com prazo de entrega apropriado.