

MC102W - Algoritmos e Programação de Computadores

Lab12: Bots trapaceiros

Prazo: 21 de Junho de 2020

Peso da Atividade: 5

Trapaça (*Cheat*, em inglês) é um jogo de baralho no qual o objetivo é descartar todas as cartas da mão. Utilizando apenas um baralho (52 cartas), o jogo permite entre 3 a 6 jogadores. Todas as cartas são distribuídas entre os jogadores. Os descartes são acumulados em uma pilha comum que poderá voltar para a mão dos jogadores em algumas condições. Em cada turno do jogo, existe uma carta alvo seguindo a ordem **A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K**. O jogador do turno deve jogar o maior número de cartas de sua mão com o mesmo número/letra do alvo (de qualquer naipe), com as faces das cartas viradas para baixo (escondidas). Essas cartas serão seu descarte do turno atual. Caso não possua nenhuma carta com o número/letra da vez, o jogador deve blefar, jogar qualquer carta e dizer que é do tipo alvo. Blefando ou não, qualquer outro jogador pode duvidar da jogada atual, gritando “Duvido!”. Caso algum adversário o faça, existem duas possibilidades. Primeiro, o jogador da vez estava falando a verdade e o jogador acusador fica com toda a pilha de descartes, incluindo o descarte do turno. Segundo, o jogador da vez estava blefando e ele mesmo fica com toda a pilha de descartes, incluindo seu descarte. Se nenhum adversário duvidar, as cartas descartadas são acumuladas na pilha.

As etapas do jogo estão detalhadas abaixo. Para o laboratório faremos uma versão simplificada do jogo que será detalhado na seção Tarefa.

1. Os jogadores sorteiam quem será o primeiro a jogar, por exemplo, pela estratégia de quem tira a carta mais alta.
2. Todas as cartas do baralho são divididas entre os jogadores. Caso a divisão não seja inteira, as remanescentes vão para a pilha de descartes.
3. O jogo começa pela carta alvo **A** no turno do primeiro jogador.
4. O jogador da vez busca em sua mão todas as cartas com o número/letra alvo, e joga as cartas com as faces viradas para baixo.
5. Os outros jogadores podem ou não duvidar que as cartas são do tipo alvo. Caso alguém duvide, o jogador da vez mostra sua jogada. Se **todas** as cartas jogadas contiverem o número/letra da vez, o acusador leva todas as cartas da pilha, inclusive o descarte atual. Se **qualquer** das cartas jogadas não for do tipo esperado, o jogador da vez leva todas as cartas da pilha, inclusive as cartas que tentou descartar.
6. Se a jogada atual teve sucesso (não houve acusação ou o jogador provou não estar blefando) e o jogador da vez não tem mais nenhuma carta na mão, ele vence. Caso contrário, um novo turno se inicia, a carta alvo muda para a próxima

da lista e o jogador da vez será o próximo no sentido horário. O jogo volta para o passo 4.

Tarefa

Sua tarefa neste laboratório é implementar um único turno do jogo de um jogador fixo (bot do turno). O objetivo é determinar qual será o descarte do bot no turno e se ele levará ou não as cartas da pilha de descartes. A entrada do programa indicará a mão atual do bot (ordenada), a pilha de descartes (em qualquer ordem) e o número/letra alvo do turno. A ordenação segue primeiro a ordem de números/letras **A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K** e depois a ordem de naipes **C, E, O, P** (C = copas, E = espadas, O = ouros, P = paus), isto é, os naipes desempatam cartas com números iguais. Por exemplo, a carta **4P** vem antes da **5E**, mas depois da carta **4C**. Sua primeira tarefa é **buscar** na mão do bot do turno cartas do tipo alvo. Se a carta for encontrada, o bot deve jogar **todas** as cartas de sua mão com o mesmo número/letra. Note que, ele irá jogar no máximo 4 cartas, uma de cada naipe. Caso a carta não seja encontrada, ele deve jogar todas as cartas que tiver do menor número/letra. A entrada também indicará se algum bot adversário duvidou da jogada atual. A sua segunda tarefa é determinar se o bot do turno leva ou não as cartas da pilha, baseado na jogada feita. Se o bot do turno não blefou (jogou somente cartas do tipo alvo), o bot adversário leva as cartas da pilha. Caso contrário, você deve adicionar as cartas da pilha à mão do bot do turno, **mantendo a mão ordenada**. Se, após a jogada, o bot do turno estiver com a mão vazia, ele vence o jogo.

Observação:

- ❑ Os métodos de ordenação e de busca devem ser implementados. **Não será permitido utilizar funções prontas.** Você deve utilizar a ordenação **Insertion Sort** e **Busca Binária** para este laboratório. Lembre-se que a mão sempre deve estar ordenada e, ao encontrar qualquer ocorrência da carta alvo, basta verificar as cartas vizinhas para retornar o maior número de cartas alvo.

O programa deve ser implementado em Python e deve utilizar métodos de ordenação e de busca.

Exemplo 1:

Mão: 4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO

Pilha: 3C AC

Carta alvo: A

Um bot adversário duvidou.

O bot do turno não tem carta **A** e joga a carta **4E** (sua menor).

Como um bot adversário duvidou, o bot do turno termina com a mão:

Mão: AC 3C 4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO

Exemplo 2:

Mão: 4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO

Pilha: 3C AC

Carta alvo: J

Um bot adversário duvidou.

O bot do turno tem carta **J**: **JC**, **JE** e **JO**.

Um bot adversário duvidou, mas o bot do turno não estava blefando. Portanto, o adversário fica com a pilha e o bot do turno termina com a mão:

Mão: 4E 6C 6E 7E 7P 8C 9O QO KO

Entrada

A entrada do programa contém:

- ☐ As cartas da mão atual do bot do turno ordenadas e separadas por espaço. A mão contém pelo menos uma carta.
- ☐ A pilha de descartes em qualquer ordem, também separadas por espaço. A pilha pode estar vazia.
- ☐ O número/letra da carta alvo.
- ☐ Uma linha indicando se algum adversário duvidou ('S') ou não ('N').

Não existem quebras de linhas entre as cartas, apenas entre os itens da entrada. As entradas são garantidamente válidas e não precisam ser verificadas. Por exemplo, a última linha sempre será 'S' ou 'N'.

Saída

A saída do programa deve conter:

- ☐ A jogada do bot do turno no formato "Jogada: <cartas>".
- ☐ Se algum bot adversário duvidou ("Um bot adversário duvidou" ou "Nenhum bot duvidou"). Se algum bot duvidou, você deve imprimir se o bot do turno blefou ou não ("O bot estava blefando" ou "O bot não estava blefando"). Lembre-se que, caso algum bot duvide do jogador e ele não tenha blefado, o adversário leva as cartas da pilha.
- ☐ A mão do bot e a pilha ao final do turno nos formatos "Mão: <cartas>" e "Pilha: <cartas>", uma por linha.
- ☐ Se a mão estiver vazia, deve indicar que o bot do turno venceu ("O bot venceu o jogo")

Observações:

- ☐ Observe que existe um espaço antes de "<cartas>" em todos os casos. Sendo assim, caso a mão/pilha esteja vazia, a impressão será "Mão/Pilha: ", i.e. com um espaço após ".".

Exemplo

Exemplo 1:

Entrada

```
4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO
3C AC
A
S
```

Saída

```
Jogada: 4E
Um bot adversário duvidou
O bot estava blefando
Mão: AC 3C 4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO
Pilha:
```

Exemplo 2:

Entrada

```
4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO
3C AC
J
S
```

Saída

```
Jogada: JC JE JO
Um bot adversário duvidou
O bot não estava blefando
Mão: 4E 6C 6E 7E 7P 8C 9O QO KO
Pilha:
```

Exemplo 3:

Entrada

```
4E 6C 6E 7E 7P 8C 9O JC JE JO QO KO
3C AC
A
N
```

Saída

```
Jogada: 4E
Nenhum bot duvidou
Mão: 6C 6E 7E 7P 8C 9O JC JE JO QO KO
Pilha: 3C AC 4E
```

Critérios específicos

Os seguintes critérios específicos sobre o envio, implementação e execução devem ser satisfeitos.

- i. Submeter no SuSy o arquivo:
⇒ `lab12.py`: Arquivo contendo todo o seu programa.
- ii. Não serão aceitas soluções contendo estruturas não vistas em sala, exceto as indicadas neste enunciado.

Observações gerais

No decorrer do semestre haverá 3 tipos de tarefas no SuSy (descritas logo abaixo). As tarefas possuirão os mesmos casos de testes abertos e fechados, no entanto o número de submissões permitidas e prazos são diferentes. As seguintes tarefas estão disponíveis no SuSy:

- ❑ **Lab12-AmbienteDeTeste**: Esta tarefa serve para testar seu programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém os arquivos submetidos aqui **não serão corrigidos**.
- ❑ **Lab12-Entrega**: Esta tarefa tem limite de uma **única** submissão e serve para entregar a **versão final** dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa e submeta aqui apenas quando não for mais fazer alterações no seu programa.
- ❑ **Lab12-ForaDoPrazo**: Esta tarefa tem limite de uma **única** submissão e serve para entregar a versão final fora prazo estabelecido para o laboratório. Esta tarefa irá substituir a nota obtida na tarefa **Lab12-Entrega** apenas se o aluno tiver realizado as correções sugeridas no *feedback* ou caso não tenha enviado anteriormente na tarefa **Lab12-Entrega**.

Avaliação

Este laboratório será avaliado da seguinte maneira: a nota será proporcional ao número de casos **fechados** para os quais o seu programa gerou a resposta correta, **desde que os critérios indicados neste enunciado tenham sido atendidos**. Se o programa apresentou

resposta correta para todos os casos, a nota será 10; caso contrário será $p \cdot 10$, onde p é o percentual de respostas corretas. A nota também poderá sofrer descontos de acordo com a qualidade do programa apresentado. Assim, mesmo que o código seja capaz de resolver todos os casos de teste fechados, a nota final ainda pode ser menor do que 10. Por isso, acrescente comentários explicativos, utilize variáveis sugestivas e faça um código claro e de acordo com o que foi solicitado.

Testando seu programa

Para testar se a solução do seu programa está correta, basta seguir o exemplo abaixo no terminal do Linux.

```
python lab12.py < arq01.in > arq01.out
diff arq01.out arq01.res
```

O `arq01.in` é a entrada e `arq01.res` é a saída esperada, ambos disponíveis no SuSy. O `arq01.out` é a saída gerada pelo seu programa. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados e testados da mesma forma que os testes abertos.