

CCM0118 - Computação I (2024ii)

[Início](#) / [Meus Ambientes](#) / [2024](#) / [RUSP](#) / [CCM](#) / [0118.2024ii](#) / [Exercícios-programa](#) / [E02 Amigo secreto](#)

E02 Amigo secreto

Vencimento: domingo, 29 set. 2024, 23:59

Parte 1. Um grupo de N amigos decide promover um amigo secreto. Para tal, eles usam um gerador de permutações aleatórias dos números $0, \dots, N - 1$. Se $a[\]$ contém a permutação gerada, eles fazem o seguinte:

- A pessoa 0 dá seu presente para a pessoa $a[0]$;
- A pessoa $a[0]$ dá seu presente para a pessoa $a[a[0]]$;
- A pessoa $a[a[0]]$ dá seu presente para a pessoa $a[a[a[0]]]$ e assim por diante.
- Se todos deram seus presentes, o processo termina. Caso contrário, uma pessoa que ainda não deu seu presente, escolhida arbitrariamente, recomeça o processo.

Por exemplo, se $N = 6$ e $a = \{3, 0, 4, 2, 5, 1\}$, então

- A pessoa 0 dá seu presente para a pessoa 3
- A pessoa 3 dá seu presente para a pessoa 2
- A pessoa 2 dá seu presente para a pessoa 4
- A pessoa 4 dá seu presente para a pessoa 5
- A pessoa 5 dá seu presente para a pessoa 1
- A pessoa 1 dá seu presente para a pessoa 0

Os N amigos executam esse plano, e percebem que a pessoa 0, que deu seu presente no começo do jogo, recebeu seu presente somente no fim do jogo, depois de todos os outros terem recebido seus presentes (como no exemplo acima). Eles ficam um tanto surpresos, e decidem investigar qual é a probabilidade p_N de acontecer tal "fenômeno S" ("S" de "surpreendente"). (A probabilidade é denotada p_N pois pode depender de N .)

Escreva um programa que estima p_N . Seu programa deve chamar-se ProbabilidadeS.java.

Mais precisamente, seu programa deve receber como entrada inteiros $NMAX$ e T . Seu programa deve ter como saída as estimativas obtidas para $p_2, p_3, \dots, p_{NMAX}$. Para obter cada p_N , seu programa deve simular o processo para N pessoas T vezes.

Exemplo de execução

```
$ java-introcs ProbabilidadeS 5 20
p_2 = 0.25
p_3 = 0.25
p_4 = 0.2
p_5 = 0.1
$
```

Parte 2. Os N amigos gostam da brincadeira e resolvem continuar suas investigações. Para o amigo secreto, é inconveniente que haja na permutação $a[\]$ um índice i tal que $a[i] = i$ (por quê?). Chamemos de *boas* as permutações que não têm tal índice i .

Seja q_N a probabilidade de uma permutação aleatória dos números $0, \dots, N - 1$ ser boa. Escreva um programa chamado ProbabilidadeBoa.java que estima q_N .

Mais precisamente, seu programa deve receber como entrada inteiros $NMAX$ e T . Seu programa deve ter como saída as estimativas obtidas para $q_2, q_3, \dots, q_{NMAX}$. Para obter cada q_N , seu programa deve gerar T permutações aleatórias no processo de estimação.

Exemplo de execução

```
$ java-introcs ProbabilidadeBoa 5 20
q_2 = 0.35
q_3 = 0.45
q_4 = 0.4
q_5 = 0.1
$
```

Pergunta. Os valores q_N parecem convergir para algum valor quando $N \rightarrow \infty$? Se você acha que este é o caso, qual é sua estimativa para esse valor?

Parte 3. Levando em conta suas descobertas, escreva um terceiro programa chamado PermutacoesBoas.java que gera permutações boas de $0, \dots, N - 1$. Note que, naturalmente, exigimos que as permutações geradas não tenham nenhum viés!





Seu programa PermutacoesBoas.java deve receber N e T na linha de comando, e deve imprimir T permutações boas de $0, \dots, N - 1$, uma por linha, como no exemplo abaixo.

Exemplo de execução

```
$ java-introcs PermutacoesBoas 5 10
0: 1 4 3 0 2
1: 3 4 0 2 1
2: 3 2 4 1 0
3: 3 0 4 2 1
4: 2 3 4 1 0
5: 3 0 1 4 2
6: 4 2 1 0 3
7: 1 3 4 2 0
8: 1 4 3 0 2
9: 4 0 1 2 3
$
```

Entrega. Você deve entregar seus programas PermutacoesBoas.java, ProbabilidadeBoa.java e ProbabilidadeS.java.

Status de envio

Status de envio	Enviado para avaliação
Status da avaliação	Avaliado
Tempo restante	A tarefa foi enviada 4 horas 14 minutos adiantado
Última modificação	domingo, 29 set. 2024, 19:44
Envios de arquivo	<div><div> PermutacoesBoas.java 29 setembro 2024, 19:44 PM</div><div> ProbabilidadeBoa.java 29 setembro 2024, 19:44 PM</div><div> ProbabilidadeS.java 29 setembro 2024, 19:44 PM</div></div>
Comentários sobre o envio	<div> Comentários (0)</div>

Feedback

Nota	10,00 / 10,00
Avaliado em	domingo, 13 out. 2024, 23:26
Avaliado por	MC Matheus Castello Branco Lima de Araujo

Comentários de feedback



Parte 1 - ProbabilidadeS:

- Gerou a permutação corretamente (2/2)
- Seguiu corretamente a ordem de presentear (1/1)
- Calculou ...

[◀ E01 Dados generalizados](#)

Seguir para...

[E03 Bolas em colisão ▶](#)