

Web Ex. 2.1.5 (Voting machines)

Faça o Web Exercise 2.1.5 (Voting Machines) de IntroCS:

<http://introcs.cs.princeton.edu/java/21function/>

(Este exercício aparece como o Creative Exercise 2.2.24 no livro; veja arquivo anexo)

Seu programa deve chamar-se `VotingMachines.java`. Em linhas gerais, seu programa deve simular a eleição descrita no enunciado várias vezes, e deve estimar a partir daí a probabilidade de o erro das urnas eletrônicas afetarem o resultado, isto é, ele deve estimar a probabilidade do resultado da votação dar mais votos para o candidato B do que para A. Sem erros, o candidato A recebe (de acordo com os dados do enunciado) 51% dos votos e ele deve ser vencedor nesta eleição. O ponto é que os erros das urnas podem afetar a contagem dos votos, resultando em mais votos para o candidato B do que para o candidato A. Queremos estimar a probabilidade desse resultado incorreto acontecer.

Seu programa deve receber como argumentos de linha de comando quatro quantidades: N , a , f e T . O inteiro N é o número de votantes (100 milhões no enunciado). O real a , entre 0 e 1, é a fração de votantes que votam no candidato A (.51 no enunciado). O real f é a probabilidade de falha das urnas eletrônicas: a probabilidade de um voto ser computado incorretamente (.05 no enunciado). Finalmente, T é um inteiro que especifica quantas vezes a votação deve ser simulada para estimarmos a probabilidade de B ser declarado vencedor.

Seja p a probabilidade de B ser declarado vencedor. Naturalmente, p depende dos parâmetros N , a e f . Assim, é natural denotar p por $p(N, a, f)$. Usando esta notação, seu programa `VotingMachines.java` estima $p(N, a, f)$.

Escreva também um programa chamado `VotingMachinesDeluxe.java`, para responder a pergunta final no enunciado. Vamos dizer que a probabilidade de falha f é inaceitável se $p(N, a, f)$ for maior que 3% e tolerável se $p(N, a, f)$ menor ou igual a 3%. Seu programa `VotingMachinesDeluxe.java` deve estimar o maior f tolerável para um dado par de parâmetros N e a .

Sugestão. Para implementar `VotingMachinesDeluxe.java`, use o método da bisseção.

Comentário do Monitor:

O enunciado acima está bem descrito, vai abaixo alguns exemplos:

O primeiro caso é o do enunciado do livro ($N = 100000000$, $a=0.51$, $f=0.05$ e $T=10000$)

```
$ java-introcs VotingMachines 100000000 0.51 0.05 5
Wrong results/trials: 0/5
These values are acceptable

$ java-introcs VotingMachines 100000000 0.51 0.5 100
Wrong results/trials: 51/100
These values are not acceptable
```

Lembrando que é aceitável uma eleição quando $p(N, a, f) \leq 3\%$, ou seja, quando $\text{numeros_resultados_errados}/\text{numero_tentativas} \leq 0.03$

Em relação ao VotingMachinesDeluxe você deve descobrir qual é o f . O seu erro deve ser menor do que 0.001, ou seja, o valor que você achar deve estar entre $]f-0.001, f+0.001[$. O programa vai receber N , a e T como argumentos.

Percebe que esses experimentos são baseados em números aleatórios, portanto os resultados do exemplo podem não coincidir, mas devem ser próximos.

No exemplo a seguir, $N=1000$, $a=0.60$ e $T=100000$

```
$ java-introcs VotingMachinesDeluxe 1000 0.60 100000
0.35947036
```

Aviso: se não for utilizar o método da bisseção, deixe bem claro o que está usando (recomendo usar o método da bisseção).