Estudo das eleições

Vamos estudar a robustez do resultado de eleições quando as urnas falham com uma certa probabilidade, mudando o voto de alguns eleitores.

Por exemplo, suponha que vai se realizar uma eleição com apenas dois candidatos, A e B, e com N = 1.000.000 votantes; não há votos brancos, nulos ou abstenções. A porcentagem de votantes no candidato A é ***a = 51%***, porém cada voto pode ser registrado com uma falha da urna, ou seja, o voto pode ser alterado para o outro candidato com probabilidade de falha ***f = 5%****.* As falhas da urna ocorrem de forma independente para cada votante.

Uma primeira pergunta natural será: qual é a probabilidade de que o candidato B obtenha mais votos que A?

Se B tiver mais votos que A, significa que a eleição teve um erro.

erro = resultado incorreto eleição;
falha = falha de uma urna na computação de um voto;

Naturalmente, a probabilidade de erro em uma eleição é uma função crescente da probabilidade de falha. Ela pode ser estimada utilizando experimentos aleatórios que simulam uma eleição ***T*** vezes: em cada simulação contamos os votos do candidato A com aquela possibilidade de falha nas urnas, e, caso ele perdesse, contamos o erro na eleição. A probabilidade de erro será estimada pela fração entre o número de eleições com ***erro*** pelo total ***T*** de simulações.

Uma outra pergunta natural é: qual é a maior probabilidade de falha de urnas para que a probabilidade de erro numa eleição seja no máximo uma certa tolerância, ***tol***?

Esta é a pergunta que seu programa deve responder, tendo em mãos:
N: o número de votantes;
a: a porcentagem de votantes no candidato A;
T: o número de simulações usadas para estimar a probabilidade de erro numa eleição;
tol: a probabilidade máxima aceita para erro numa eleição.

Exemplos de entrada e saída 1

Digite o número de votantes (0 < N <= 2x10^9): 100

Digite a porcentagem de votos do candidato A (0.5 < a <= 1): .55

Digite o número de simulações (0 < T <= 2x10^9): 10000

Digite a probabilidade tolerável de erros (0 <= tol <= 1): .01

Maior falha das urnas tolerável: 0.0470386

Exemplos de entrada e saída 2

Digite o número de votantes (0 < N <= 2x10^9): 1000

Digite a porcentagem de votos do candidato A (0.5 < a <= 1): .51

Digite o número de simulações (0 < T <= 2x10^9): 10000

Digite a probabilidade tolerável de erros (0 <= tol <= 1): .01

Maior falha das urnas tolerável: 0.0197148

Seu programa deve obrigatoriamente começar com os itens dados abaixo:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define BISSEC_TOL (1e-6)

/* "dá a partida" no gerador de números aleatórios */
void ativa_sorteador()
{
#ifdef RANDOM_SEED
srand(RANDOM_SEED);
#else
  srand(time(NULL));
#endif
}

/* devolve um real sorteado uniformemente no intervalo [0,1] */
double sorteia_real()
{
return (double) rand() / RAND_MAX; }
```

Logo após a declaração de variáveis na função **main**, você deve fazer a chamada de função: ativa_sorteador();

<u>TAREFA:</u>

Além de criar a **main**, você necessariamente deve implementar as seguintes funções com estes protótipos e especificações:

• **int sorteia_voto_com_falha (double f):** *devolve um inteiro não nulo com probabilidade f, e devolve 0 com probabilidade 1 − f, onde 0 ≤ f ≤ 1.*

• **double prob_erro (int N, double a, double f, int T):** *estima a probabilidade de erro de uma eleição com N votantes, dos quais uma fração* ***a*** *vota no candidato A, e com probabilidade de falha f, utilizando T simulações. Os limites dos parâmetros são como no exemplo de entrada e saída.*

double bissecta (int N, double a, int T, double tol): *calcula a probabilidade máxima de falha das urnas para que a probabilidade de erro de uma eleição seja limitada superiormente por tol, utilizando o método da bissecção, como descrito a seguir.*

<u>O método da bissecção</u>

No nosso caso, queremos estimar a maior probabilidade de falha que ainda garante a vitória de A na eleição com probabilidade pelo menos *1 − tol* (lembre que estamos supondo que uma fração maior que 50% dos votantes vota em A, e que queremos que a probabilidade de a eleição resultar em erro seja de no máximo *tol*).

Observe que sabemos que com probabilidade $f = f_a = 0$ (não há falhas), certamente A ganhará a eleição. Da mesma forma, se a probabilidade for $f = f_b = 1$, ocorrerá erro, e B vencerá de forma incorreta. A ideia do método é, com posse destas duas probabilidades f_a e f_b, olhar o que acontece no ponto médio dos dois valores. Se neste ponto A ganha a eleição com probabilidade pelo menos *1 − tol*, podemos atualizar f_a. Por outro lado, se neste ponto a chance de vitória de B supera a tolerância *tol*, atualizamos f_b. Este processo é repetido até que a diferença de f_a e f_b seja menor que o valor BISEC_TOL.