



Universidade Federal do Ceará
Disciplina: Matemática e Física para Jogos
Professor: Gilvan Maia
Aluno: Ubiratan Júnior
LISTA DE EXERCÍCIOS 02 - Unidade 01

1. Pesquise pelos termos CRC (*Cyclic Redundance Check*) e *Checksum*

a) O que é CRC e quais são as suas aplicações?

É uma técnica de detecção de erros que normalmente é usada em meios de comunicação sem fio, a técnica consiste em adicionar um valor ao arquivo baseado no resto de divisão polinomial do seu conteúdo. No ato da recuperação do dado o cálculo é refeito e comparado com o valor gerado anteriormente.

b) Os algoritmos de CRC e checksum são inteiramente confiáveis?

Não, o checksum por se tratar de uma soma, é vulnerável diante da propriedade de comutatividade da soma enquanto o CRC não é aplicável para proteção contra alteração intencional dos dados pois não há nenhum tipo de autenticação, logo um invasor pode editar a mensagem e recalcular seu código de verificação sem que a substituição seja detectada.

2. Pesquise sobre Geradores de Números Aleatórios

a) Esses números são realmente aleatórios?

Não, como os números são gerados por computadores, eles não tem a capacidade de executar experimentos totalmente aleatórios por si mesmo, a alternativa para se gerar esse tipo de número é usar informações de um evento externo que sejam passadas para a máquina.

b) Cite exemplos de situações nas quais não é desejável ter um gerador totalmente aleatório.

Um caso onde é desejável controlar a aleatoriedade seria quando desejamos compensar diferença de nível em jogos multiplayer online que tem equipes geradas aleatoriamente, por exemplo em filas ranqueadas de FPS ou Mobas, se as equipes forem formadas de forma totalmente aleatória há a chance de os níveis dos times formados ser muito desigual.

c) O que é "semente" (*seed*) no contexto dos geradores de números pseudoaleatórios?

Como os geradores não são totalmente aleatórios, eles partem de um ponto de inicial e determinístico para iniciar a gerar os números, o seed é exatamente esse ponto.

d) Quais algoritmos para geração de números aleatórios estão disponíveis na sua linguagem de programação favorita?

Em JavaScript nativamente temos apenas a função `Math.random()`, um fato interessante é que não há a possibilidade de alterar o seed dessa função segundo a documentação encontrada em MDN web docs

3. Já ouviu falar do lendário código para calcular rapidamente o inverso multiplicativo da raiz quadrada? Pesquise sobre a função *InvSqrt()*. Como é possível estimar $1.0/\text{sqrt}(x)$ sem recorrer a divisões nem à raiz quadrada ?

Nunca tinha visto falar sobre a função, o cálculo sem divisões e uso de raízes quadradas só é possível por conta da constante *0x5f3759df* e do operador *>>* (que nada mais é que uma forma "mecânica" de se efetuar uma divisão por 2).

4. Pesquise sobre as séries de Taylor

- a) **Como utilizar séries de Taylor para calcular o valor aproximado de funções trigonométricas como *seno*, *cosseno* e *tangente*?**

Como as funções trigonométricas podem ser derivadas n vezes podemos simplesmente usar a fórmula:

$$p(x) = f(a) + f'(a)\frac{(x-a)^1}{1!} + f''(a)\frac{(x-a)^2}{2!} + \dots + f^{(n)}(a)\frac{(x-a)^n}{n!}$$

Substituindo f pela função desejada.

- b) **Você consegue enxergar a importância desse artifício teórico no cálculo de raízes de funções, nos problemas de otimização (e.g, estimar o valor mínimo ou máximo de uma função), nos jogos eletrônicos e na Computação Gráfica?**

Sim, visto que o uso de funções mais simples para solucionar funções complicadas através de aproximações com erro baixo são fundamentais para garantir que um algoritmo tenha ótimo desempenho e entregue o máximo de fidelidade/corretude nos cálculos.