

# Laboratório de Algoritmos I – Geração de números aleatórios

Thiago Silva Vilela

# Números Aleatórios

- Números aleatórios são importantes em todas as áreas da computação:
  - Jogos: podem ser usados, por exemplo, como localização inicial de inimigos;
  - Inteligência artificial: usados para ajudar a definir comportamentos de agentes;
  - Podem ser usados para fazer simulações de carga de websites, de fenômenos físicos, etc.
- Gerar números verdadeiramente aleatórios não é fácil.
- A maioria dos geradores de números aleatórios geram, na verdade, números pseudo-aleatórios.
  - os números são gerados através de um algoritmo.

# Função rand()

- Esta função retorna um número inteiro pseudo-aleatório (sorteado). Ou seja, no trecho de código abaixo, um valor pseudo-aleatório será atribuído à variável **a**:

```
a = rand( );
```

# Função rand()

- Em alguns casos, pode-se querer limitar o intervalo de valores possíveis para o “sorteio” do número. Normalmente o que é feito nestes casos é o uso do operador “módulo” ou “resto da divisão”. Ou seja, no trecho de código abaixo, um valor pseudo-aleatório entre 0 e 9 será atribuído à variável **a**:

```
a = rand( ) % 10;
```

# Função rand()

- Quando se quer alterar o início do intervalo dos números, deve-se somar o valor necessário para isso. Exemplo: se ao invés de 0 a 9, o intervalo for de 1 a 10, basta somar 1 ao resultado anterior:

```
a = rand() % 10 + 1;
```

# Função srand()

- A função srand() faz com que a sequência de números pseudo-aleatórios gerados pela função rand() seja diferente, de acordo com a semente que é passada para ela. Para isso, deve-se passar um valor para srand(), e sempre que este mesmo valor for passado, a mesma sequência de valores será gerada.

Por exemplo:

```
srand( 5 );
```

Define a semente da função rand() como o número 5.

# Função srand()

- Quando se quer que os números gerados sejam realmente aleatórios, deve-se passar um número diferente para srand() a cada nova execução do programa.
- O que é feito neste caso é passar a data/hora atual do sistema para a função srand(), ou seja, sempre será diferente. Para isso, precisamos usar a diretiva `#include <time.h>` no começo do programa.
- Usamos, então, a função srand() da seguinte forma:

```
srand( time( NULL ) );
```