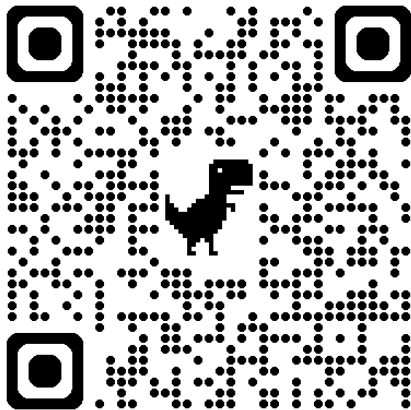# Application: 1A2B

Tai, Wei Hsuan

week 8

Announcement

Recording or photographing the slides or any content displayed on the screen is permitted for personal use only.

Redistribution, modification, or any commercial use of the materials is strictly prohibited.

Outline

Warn Up: Calculate Pi with Monte Carlo method

What is Pi?

Pi (π) is a mathematical constant that represents the ratio of a circle's circumference to its diameter. It is an irrational number, meaning it cannot be expressed as a simple fraction, and its decimal representation goes on infinitely without repeating. The value of π is approximately 3.14159.

If a circle has a radius of r, then its area would be

$$A = \pi r^2.$$

Monte Carlo method

Intuitively, if we drop random points into a square that encloses a quarter circle, the ratio of points that fall inside the quarter circle to the total number of points should approximate the ratio of the areas of the quarter circle to the square.

How to generate random numbers?

In C++, we can leverage the <random> library to generate random numbers. With rand() function, we can generate a random integer between 0 and RAND_MAX. We can use remainder operation to limit the range. However, there may be some issue with this function.

- Range is too small (i.e., RAND_MAX is at least 32767)
- Not thread-safe, people can estimate the next number.
- Not uniformly distributed.
- Remainder operation may introduce bias.

Mt19937 is a widely used pseudorandom number generator (PRNG) known for its high quality and long period. It is based on the Mersenne Twister algorithm, which was developed by Makoto Matsumoto and Takuji Nishimura in 1997. The "19937" in its name refers to the fact that it has a period of $2^{19937} - 1$, meaning it can generate a vast sequence of random numbers before repeating.

Usage

```
1  random_device rd;  // Obtain a random number from hardware
2  mt19937 eng(rd()); // Seed the generator
3  uniform_int_distribution<int>(a,b) rand_int(a, b); // Define the
   ↳  range
4  uniform_real_distribution<double> rand_real(0.0, 1.0); // Define the
   ↳  range
5  double random_value = rand_real(eng); // Generate random number
```

```cpp
#include<bits/stdc++.h>
using namespace std;

random_device rd;
mt19937 g(rd());
uniform_int_distribution<int> dist(0, 1000000);

int main(){
    cout<<dist(g)<<"\n";
}
```

Overview

What is 1A2B?

1A2B is a number guessing game. The classic version is: the computer randomly generates a 4-digit number with no repeated digits, and the player tries to guess it. After each guess, the computer provides feedback in the form of "A" and "B". A indicates the number of digits that are correct and in the correct position, while B indicates the number of correct digits but in the wrong position. The game continues until the player guesses the correct number (4A0B).

Game Process

1. Computer generates answer.
2. Player makes a guess.
3. Computer provides feedback in the form of "A" and "B".
4. Repeat steps 2 and 3 until the player guesses the correct number.

How to generate an answer?

The most important part of the game is to generate a random 4-digit number with no repeated digits.