과목 명: 고급 소프트웨어 실습

담당 교수 명: 박 운 상

<<과제 1>>

**서강대학교 컴퓨터학과**

**[20171693]**

**[조병화]**

목 차

1. 메르센 트위스터 난수 생성기를 이용하여 원주율 구하기
2. Linear congruential
3. 메르센 트위스터
4. 기타 난수 생성 방법
5. 메르센 트위스터 난수 생성 방법을 사용하여 원주율 구하기

이전 실습에서 rand()함수를 이용하여 원주율을 구해보면 오차율이 1퍼센트 이상으로 나오는 경우들 도 있었다. 이를 줄이기 위하여 기존 rand()함수보다 더 질 높은 난수를 생성해주는 메르센 트위스터 난수 생성 방법을 이용해 보았다. C++ 11에 랜덤 생성기가 포함이 되어서 사용할 수 있다.

1) std::random\_device rd;

2) std::mt19937 gen(rd());

3) std::uniform\_real\_distribution<> dis(-100.00, 100.00);

위 처럼 엔진을 선언하고, distribution 의 타입과 데이터 범위를 설정해준다. 이 때 나 같은 경우 처음에 uniform\_int\_distribution으로 선언하고 진행하였더니 오차율이 가끔 1퍼센트를 넘기는 문제가 생겨 real로 바꿔주어 생성되는 난수를 실수로 바꿔주었다.

1. Linear congruential 난수 생성 방법

Linear congruential 난수 생성 방법은 선형합동법이라고 부르기도 한다. 품질은 분포가 치우쳐져 있고 현재 값을 알 때 이후의 값을 예측할 수 있어 취약해 좋지않다고 알려져 있다. 그러나 계산속도가 빠르기 때문에 난수 품질이 그렇게 중요하지 않은 경우에 많이 사용된다고 알려져 있다.

Linear congruential 은 다음의 재귀 관계로 된 순열을 통해 난수를 생성한다.

Xi+1 = (aXi +c)mod m, I = 0,1,2…

예를 들어 X0 이 27, a를 17, c를 43, m을 100으로 두었다 하면,

Xi 과 Ri 의 값은 다음과 같이 나타난다.

X1 = (17×27+43) mod 100 = 502 mod 100 = 2 -> R1 = 0.02

X2 = (17×2 +43) mod 100 = 77 -> R2 = 0.77

X3 = (17×77+43) mod 100 = 52-> R3 = 0.52

X4 = (17×52+43) mod 100 = 27 Æ R3 = 0.27 다음 처럼 난수가 생성된다.

여기서 더 발전된 형태의 combined linear congruential generator 또한 존재한다고 한다.

1. 메르센 트위스터 난수 생성 방법

메르센 트위스터 난수 생성 방법은 그 난수의 반복 주기가 메르센 소수인 방법이다. 이 난수의 주기가 매우 크기 때문에 겹치는 난수가 생성이 잘 되지 않고, 623차원까지 동일분포되어 있다. 추가적으로 비트 연산만으로 알고리즘의 구현이 가능하기 때문에 매우 빠르다는 장점도 가지고 있다. 하지만, 생성기의 상태가 꽤 큰 편이기 때문에 적은 메모리를 사용하는 환경에서 적합하지 않을 수 있다. 암호학적으로 안전하게 설계되어 있지도 않다고 한다.

간단한 작동 원리는 다음과 같다.

첫째로, seed에서 624 만큼의 크기의 메르센 트위스터 벡터(MT벡터)를 생성하고,

둘째로 그 MT벡터를 이용하여 624개의 유사 난수를 발생시킨다.

마지막으로 MT 벡터에 노이즈를 준 후(twist) 두번 째 과정부터 반복하도록 한다.

1. 기타 난수 생성 방법
2. 정규 분포(normal distribution)

텍스트이(가) 표시된 사진

자동 생성된 설명

c++의 <random>라이브러리에서 제공하는 난수 생성 방법 중 한가지로,

std::normal\_distribution<> dis(0, 100.00);

아까의 메르센 트위스트와 비슷한 방식으로 범위 대신 평균과 편차를 적어주었다. 그리고 그 결과는 위 사진과 같다. 이 난수 생성법은 값이 평균값 주변에 많이 있는 난수들이 필요할 때 사용하면 유용하다.

텍스트이(가) 표시된 사진

자동 생성된 설명

정규분포 난수 생성 코드

1. binomial\_distribution

마찬가지로 c++ 랜덤 라이브러리 에서 제공하는 것이다.

사용법과 그 결과는 다음과 같이 나온다. 음수범위를 지정하는 법을 몰라서 실패한 것 같다..

