

과제 2

[알고리즘 01분반]

소프트웨어학부 20223908학번 김주영

개발환경

IDE: CLion, 언어: C++17, 컴파일러: g++(권장), MSVC

주어진 매개변수에 대한 분석

Round, numbers[], previousBet[]를 매개변수로 받는다. 각각에 대한 정보를 분석해보고자 한다.

먼저 round는 main함수의 반복문에 정의되었고 이를 playground에서 전달받고 decideBet에 전달한다. 이는 현재 라운드 횟수를 의미한다.

다음으로 numbers[] 또한 main함수에 정의한 배열이다. 이는 참가자가 배팅할 수 있는 숫자를 저장한다. 주어진 파일에서는 10, 20, 30, 40, 50이다.

PreviousBet 배열은 이전 배팅 결과에 대한 정보를 가진다. 이 또한 main함수에서 정의하고 있으며 크기는 1차원으로 ARRAY_SIZE로 정해져 있다. 바로 직전의 결과만을 1차원 배열로써 저장한다. 각 원소는 몇 명의 참가자가 선택했는지에 대한 내용이 저장 되어있다.

알고리즘

알고리즘을 고안하기에 앞서 몇 가지 가정을 세웠다.

i) 대부분의 다른 Player는 previousBet배열을 통해 직전 라운드 결과에 의존한 알고리즘을 바탕으로 배팅한다.

ii) 고정된 상수로써 배팅하는 경우는 없다.

위의 가정은 충분히 합리적인 알고리즘이며 주어진 정보를 가장 직관적으로 활용한 예이다.

본 과제는 다른 Player과 경쟁을 하므로 대부분이 작성했을 위 가정과 같은 알고리즘을 겨냥한 알고리즘을 고안하였다.

직전 라운드에서의 각 원소에 대한 가중치를 비교했을 때, 가장 높은 가중치의 원소를

많은 Player들이 선택했을 것이므로 바로 다음으로 높은 가중치를 가지는 원소를 선택한다. 여기서 가중치는 (배팅한 숫자)/(선택한 사람의 수)로써 정의한다.

예외처리

선택할 수 있는 숫자가 1개 이하인 경우, 두 번째로 가중치가 큰 원소를 선택할 수 없으므로 이 부분에 대한 예외를 처리하였다.

가중치((배팅한 숫자)/(선택한 사람의 수))를 구하는 과정에서 (선택한 사람의 수)가 0인 경우 런타임 에러가 발생한다. 따라서 가중치를 구하는 과정에서 (선택한 사람의 수)에 1을 더해 계산한다. 분모에 1을 더하면 결과 자체는 달라지나 원소들의 대소관계는 여전히 성립한다.

결과

```
Player 1 score: 1634
Player 2 score: 1627
Player 3 score: 2933
Player 4 score: 3562
Player 5 score: 2132
```

위와 같이 정상적으로 동작하는 것을 확인할 수 있다. 결과가 좋은 편이 아니다. 이는 주어진 파일에서 다른 Player는 배팅을 무작위로 하므로 '알고리즘 고안'에서 언급한 가정이 성립하지 않기 때문이다.