<Data Structures Final Project Summary>

저희 조는 '**수업시간에 배웠던 자료구조를 어떻게 효과적으로 주어진 데이터에 사용할지**' 에 대해 초점을 맞추고 프로젝트를 진행하였습니다.

데이터는 정렬되지 않은 상태였고 '**어느 시점에, 어떤 필요성에 의해, 이러한 자료구조가 필요하겠다**' 라는 기본생각을 바탕으로 클라이언트(고객)에게 유의미한 정보를 제공해 줄 task 문항을 고민한 뒤, 다음과 같은 프로그램을 구현하였습니다.

<투자 컨설팅 프로그램>

1. 특정한 범위의 Risk을 보고싶습니까?
2. 특정 시나리오 번호를 통해 각 종목의 투자비율을 보고싶습니까?
3. 위의 시나리오 번호에서 투자비율이 높은 종목을 보고싶습니까?
4. 특정 Risk에서, 특정 투자비율 이상을 가진 종목이 궁금하십니까?
5. 특정 종목이 포함되었을 때, Risk값이 궁금하십니까?
6. 특정 종목이, 특정 투자비율 이상을 가질 때의 Risk값이 궁금하십니까?
7. 문제를 푸는데 걸린 시간과 Risk 간의 관계가 궁금하십니까?
8. 아무것도 궁금하지 않으신가요?

문항 별로 간단히 구현 원리와 활용한 자료구조를 설명하자면 다음과 같습니다.

1. 특정한 범위의 Risk을 보고싶습니까?

-> Risk값을 SelectionSort 및 MergeSort를 통해 오름차순으로 효과적으로 정렬한 뒤 Array를 만들어 담고 있다가 for문을 활용하여 입력 받은 특정 범위에서의 Risk값을 출력하였습니다.

2. 특정 시나리오 번호를 통해 각 종목의 투자비율을 보고싶습니까?

-> 시나리오 번호와 해당 행의 투자비율을 Hashmap으로 저장하여 키 값인 시나리오 번호를 통해 value로 투자비율을 쉽게 출력할 수 있게 구현하였습니다.

3. 위의 시나리오 번호에서 투자비율이 높은 종목을 보고싶습니까?

-> 투자비율은 정렬이 되어있지 않은 상태였는데 Mergesort로 정렬을 시켜준 뒤, 오름차순으로 정렬된 것을 Stack의 LIFO 특성을 이용하여 가장 큰 값의 투자비율을 나중에 push 해준 뒤 pop 시켜서 다시 내림차순으로 정렬시켰습니다.

4. 특정 Risk에서, 특정 투자비율 이상을 가진 종목이 궁금하십니까?

-> 앞서만든 Hashmap 에서 리스크를 전달해서 시나리오를 받은 뒤 이들의 투자비율을 ArrayList에 저장시켰습니다. 이 후, 특정 투자비율보다 큰 값들을 Queue의 LIFO 특성을 이용하여 add, poll 시켜서 출력하였습니다.

5. 특정 종목이 포함되었을 때, Risk값이 궁금하십니까?

-> 커서 역할을 하는 Iterator가 시나리오에 접근하였고 Risk값을 TreeSet을 활용해 생성한 tree객체에 저장시켜 값이 추가될 때 마다 정렬이 될 수 있도록 구현하여 검색과 정렬에 효과적일 수 있게 의도하였습니다. 이 후 정렬된 Risk값을 출력하였습니다.

6. 특정 종목이, 특정 투자비율 이상을 가질 때의 Risk값이 궁금하십니까?

-> 동일하게 TreeSet과 Iterator를 사용하여 투자비율까지 접근을 하였고 이 후, 특정 투자비율보다 큰 투자비율을 가졌을 때의 Risk값과 해당 시나리오를 출력하였습니다. 투자비율과 리스값에 접근할 때에는 앞서 Hashmap을 활용하여 생성한 객체를 통해 접근하였습니다.

7. 문제를 푸는데 걸린 시간과 Risk 간의 관계가 궁금하십니까?

-> 투자자 입장에서 단기 투자의 인사이트를 원하는 경우가 생길 것을 대비하여, 문제를 푸는데 걸린 시간을 별도의 텍스트로 저장하였고 Hashmap을 사용하여 걸린 시간을 키 값으로, 해당 시나리오 번호를 Value값으로 설정하여 접근을 용이하게 한 뒤, 역시 ArrayList와 TreeSet을 활용하여 인덱스 접근과 투자비율 정렬을 효과적으로 진행하였습니다.

8. 아무것도 궁금하지 않으신가요?

-> 프로그램을 exit(종료)하기 위한 문항입니다.

이 후, 분석한 데이터의 신뢰성 확보를 위해 추가적으로 파이썬에서 실제 자바에서 분석한 값이 정확한지 확인 절차를 거쳤고, 필요한 정보를 그래프로서 시각화하기 위해 파이썬 내 matplotlib 라이브러리를 효과적으로 표현해보았습니다.

<논문이 전달하는 바>

논문을 최대한 해석해보려 노력한 결과, 이번 과제로 나온 Risk optimization problem의 이론적 기반이 되는 논문이 Iscoe와 그 연구진들이 2009년에 발표한 논문 임을 알게 되었습니다.

Iscoe의 논문을 살펴보면 credit loss distribution을 approximation 할 수 있는 3가지의 방법론이 나오는데, 각각은 Monte Carlo sampling 사용과 Central Limit Theorem사용 그리고 Law of Large Number의 사용이었습니다.

그리고 이 논문에서 기반으로 다룬 model 은 Structural models (Gupton,1997; Iscoe 1999) 임도 확인하였습니다. 이를 통해 각 approximation이 유의미 하다는 것을 밝혔고, 결과적으로 각 approximation의 사용이 상황에 따라 다르게 사용되는 것이 좋다는 것 또한 밝혔음을 확인하였습니다. Iscoe의 논문에서는 Monte Carlo sampling은 많은 양의 scenario가 있고 linear한 경우에 유의미한 loss distribution approximation이 가능하다고 밝혔습니다.

여기서 loss distribution의 approximation이 왜 중요하냐면, model의 설정과 샘플을 바탕으로 loss distribution(F^)의 근사가 선행되어야 Risk Measure을 할 수 있기 때문입니다. 문제 논문에서 CVaR(conditional Value at Rist)라고 표시된 부분은 Iscoe논문에서 ES(Expected Shortfall)과 같은 의미를 가졌는데, Iscoe논문에서는 model의 설정과 loss distribution approximation이 끝난 후 이를 바탕으로 Risk Measure를 해냈음을 알 수 있습니다.

그리고 이것의 유의미성이 선행되었기에 문제논문에서 많은 scenario가 주어지고 linear한 환경을 주어 Monte Carlo sampling을 통한 결과물로 Risk Optimization이 가능했다는 사실을 이해하게 되었습니다.