<데이터 구조 및 알고리즘 시각화>

For < 데이터 구조 및 알고리즘 시각화 시스템>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **<2024/03/30>** | **<1.0>** | **<문서생성>** | **<황의현>** |
| **<2024/05/06>** | **<1.0>** | **<문서 수정>** | **<황의현>** |
| **<2024/05/12>** | **<0.1>** | **<앱 작성 내역 추가>** | **<황의현>** |
|  |  |  |  |

목차

1. 소개 4

1.1 목적 4

1.2 범위 4

2. 요구사항 4

2.1 기능적 요구사항 4

2.2 비기능적 요구사항 4

3. 시스템 구조 4

3.1 사용자 인터페이스 4

3.2 데이터 구조 및 알고리즘 시각화 엔진 4

4. 인터페이스 4

5. 기타 요구사항 5

6. 결론 5

7. 앱 작성 내역 5

7.1 MainActivity.java 5

7.2 SortingView.java 5

7.3 activity\_main.xml 5

# 소개

## 목적

이 문서의 목적은 데이터 구조와 알고리즘을 시각적으로 보여주는 안드로이드 애플리케이션의 요구사항을 정의하는 것입니다.

## 범위

이 애플리케이션은 사용자가 다양한 데이터 구조와 알고리즘을 선택하고 시각화하여 학습할 수 있는 기능을 제공합니다

# 요구사항

## 기능적 요구사항

사용자는 앱을 실행하고 데이터 구조 및 알고리즘 목록을 볼 수 있어야 합니다.

사용자는 선택한 데이터 구조나 알고리즘을 시각화하여 볼 수 있어야 합니다.

앱은 배열, 연결 리스트, 스택, 큐, 트리, 그래프 등 다양한 데이터 구조를 지원해야 합니다.

앱은 정렬, 검색, 그래프 탐색 등 다양한 알고리즘을 시각화할 수 있어야 합니다.

사용자는 각 시각화된 데이터 구조나 알고리즘에 대한 설명을 볼 수 있어야 합니다.

## 비기능적 요구사항

앱은 직관적이고 사용하기 쉬운 사용자 인터페이스를 제공해야 합니다.

시각화는 성능에 영향을 주지 않아야 하며, 부드럽고 빠른 사용자 경험을 제공해야 합니다.

앱은 안정적이고 오류 없이 작동해야 하며, 예외 상황에 대한 적절한 처리가 이루어져야 합니다.

시각화된 데이터 구조나 알고리즘의 설명은 정확하고 명확해야 합니다.

# 시스템 구조

## 사용자 인터페이스

앱은 메인 화면에 데이터 구조와 알고리즘 목록을 표시합니다.

사용자는 목록에서 항목을 선택하여 시각화를 시작할 수 있습니다.

## 데이터 구조 및 알고리즘 시각화 엔진

앱은 선택한 데이터 구조나 알고리즘을 시각화하기 위한 엔진을 포함합니다.

이 엔진은 사용자의 입력을 받아 시각화된 결과를 생성합니다.

# 인터페이스

메인 화면: 데이터 구조와 알고리즘 목록을 표시하는 화면

시각화 화면: 선택한 데이터 구조나 알고리즘을 시각화하는 화면

# 기타 요구사항

앱은 오프라인에서도 사용할 수 있어야 합니다.

사용자는 시각화된 결과를 공유할 수 있는 기능을 제공받아야 합니다.

# 결론

이 데이터 구조 및 알고리즘 시각화 안드로이드 애플리케이션은 사용자가 학습하고 이해할 수 있는 다양한 데이터 구조와 알고리즘을 제공합니다. 사용자는 직관적이고 사용하기 쉬운 인터페이스를 통해 앱을 쉽게 탐색하고, 선택한 데이터 구조 및 알고리즘을 시각화하여 확인할 수 있습니다. 이를 통해 사용자는 오프라인이거나 온라인에서 모바일 디바이스를 통해 언제 어디서든 데이터 구조와 알고리즘을 학습하고 공유할 수 있습니다.

참고로, 시각화된 결과와 각 데이터 구조 및 알고리즘에 대한 명확하고 이해하기 쉬운 설명은 사용자가 학습을 도와줄 것입니다. 이러한 요구사항을 충족시키는 안드로이드 애플리케이션을 개발함으로써 사용자들에게 탁월한 학습 경험을 제공할 것입니다.

# 앱 작성 내역

## MainActivity.java

안드로이드 애플리케이션의 메인 액티비티를 구현했습니다.

사용자가 입력한 숫자를 받아와 배열로 변환하고, 그 배열을 정렬하여 정렬된 결과를 화면에 표시하는 기능을 구현했습니다.

UI 구성 요소와 액티비티 생명주기를 관리하는 코드를 작성했습니다.

## SortingView.java

데이터를 시각적으로 표현하기 위한 커스텀 뷰인 SortingView 클래스를 구현했습니다.

입력된 배열에 따라 막대 그래프를 그려주는 기능을 구현했습니다.

배열의 최대값을 계산하여 막대의 높이를 비례하여 조정하는 로직을 작성했습니다.

## activity\_main.xml

MainActivity의 레이아웃을 XML로 정의했습니다.

EditText, Button, TextView 등의 UI 요소를 배치하고 각 요소에 대한 ID를 설정했습니다.

액티비티의 레이아웃을 화면에 표시하는 데 필요한 XML 코드를 작성했습니다.