

河內塔問題系統規劃文件

目錄

1. 簡介
2. 問題描述
3. 解決方案概述
4. 系統架構
5. 模組說明
6. 介面定義
7. 測試計劃
8. 開發環境
9. 風險評估
10. 參考資料

1. 簡介

本文件旨在定義河內塔問題的解決方案，提供開發人員清晰的思路和指導，以實現一個有效的河內塔求解系統。

2. 問題描述

河內塔問題的目標是將一堆不同大小的圓盤從「三個塔」中起始塔上移動到目標塔上，並遵循以下規則：

- 每次只能移動一個圓盤。
- 大圓盤不能疊在小圓盤之上。

3. 解決方案概述

我們將開發一個河內塔求解系統，該系統將根據輸入的圓盤數目，計算出移動圓盤的步驟，並提供一個漸進式的解決方法。我們將使用「遞迴」的技巧來解決河內塔問題。核心思想是將大問題分解為更小的子問題，然後遞迴地解決這些子問題，最終組合成大問題的解。

大問題：將 n 個圓盤依河內塔規則由大到小排序到目標塔上。

子問題：

1. 移動 $n-1$ 個圓盤從起始針到暫存塔。
2. 移動最大的圓盤從起始針到目標塔。
3. 移動 $n-1$ 個圓盤從輔助針到目標塔。

第一個子問題會在拆解成更多子問題。

子問題的「大問題」：將 $n-1$ 個圓盤依河內塔規則由大到小排序到子問題的「目標塔」上。

子問題的「子問題」：

- 移動 $n-2$ 個圓盤從起始針到暫存塔。
- 移動最大的圓盤從起始針到目標塔。
- 移動 $n-2$ 個圓盤從輔助針到目標塔。

...一直延伸子問題，

最後將最小的圓盤放到目標塔上。

4. 系統架構

前端：

河內塔求解系統只由前端 **React** 完成，將包含以下組件及功能：

- **Header 組件**
- **遞迴功能**
- **結果顯示模組：**
 - List 組件
 - Item 組件

後端：

- **Node.js server**

5. 模組說明

Header 組件：

1. 負責接收使用者輸入的圓盤數目。
2. 確認輸入參數的合法性，避免無效輸入。
3. 初始化河內塔的初始狀態，將圓盤放置在起始塔上。

遞迴功能：

1. 實現遞迴算法，解決河內塔問題。
2. 記錄每一步的移動過程，以便後續顯示。

結果顯示模組：

List 組件：顯示每個移動步驟的 **Item** 組件列表。

Item 組件：顯示特定移動步驟的詳細資訊。

Node.js 伺服器：

負責提供前端所需的公共資源，例如 **React** 程式碼、**CSS** 樣式和圖片等。

6. 介面定義

1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2		2													
3		3	3	3		1	1		1			1	1		2
4		4		4	4	1	4	2	2	1	2	2	3	2	3
5		5	1	5	2	1	5	2	3	5	2	3	5	4	1
A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
2	2	2		1	1			1	1		1	1		3
3	3	3	3	2	2	2	1	2	2	4	2	4	1	4
5	4	4	5	1	4	5	1	4	5	3	4	5	3	2
A	B	C	A	B	C	A	B	C	A	B	C	A	B	C

31	32
2	2
3	3
4	4
1	5
A	B

7. 測試計劃

1. 測試小規模盤數
2. 測試中等規模盤數
3. 測試大規模盤數
4. 測試輸入非法參數的情況

8. 開發環境

- 前端：React
- 後端：Node.js

9. 風險評估

- 遞迴可能導致堆疊溢位。

10. 參考資料

- 河內塔問題相關文獻和資源。