Stacking Boxes

99753506 林明慶
98703054 林暐智
98703018 莊士鋒

Agenda

- 103 – Stacking Boxes Problem
- Methods
- Analysis

103 – Stacking Boxes Problem(1/4)

- 在數學和計算機學中，一維或二維的概念是比較簡單的。
- 當維數不確定時，則就會變得比較複雜。
- 比方說，求**高階微分方程式**和**分析n維的超立方體拓樸結構**中，高階微分方程式要比維數為一階要複雜得多，而n維的超立方體拓樸結構則與較低階的同類具有非常大之相似之處。

103 – Stacking Boxes Problem(2/4)

- 給定一個n維"盒子"的維數。
- 在二維之情況下，box(2,3)可以表示一個長2個單位，寬3個單位的盒子。
- 在三維情況下，box(4,8,9)可以表示一個4 x 8 x 9（長寬高）的盒子。
- 在六維空間中，可能無法搞清楚box(4,5,6,7,8,9)是甚麼樣子，但我們可以分析盒子的性質，例如:求其維度之和。

103 – Stacking Boxes Problem(3/4)

- 要確定出盒子的最長嵌套串。也就是說一系列的盒子b_1、b_2、……、b_k，一個套一個，使所有的b_i都嵌套在b_{i+1}內。
- 對於一個盒子D=(d_1, d_2, ..., d_n)和另一個盒子E=(e_1, e_2, ..., e_n)，如果存在一種d_i的排列，使重排的每個維度的值都小於盒子E中對應維度的值，則盒子D可嵌入盒子E。這個過程和旋轉盒子D，看它是否能套入E的過程類似。
- 然而，若需滿足所有維度的排列，盒子D必需是可以扭曲的，而不僅僅是旋轉。

103 – Stacking Boxes Problem(4/4)

- 例如:
  1. 盒子D=(2,6)可嵌入盒子E=(7,3)是由於盒子可以排列為(6,2)，使每個維度都小於E中的相應維度。
  2. 而盒子D=(9,5,7,3)就不能嵌入盒子E=(2,10,6,8)，因為不存在D的一種排列可以滿足嵌入的性質，但是F=(9,5,7,1)就可以通過排列為(1,9,5,7)而嵌入E。
- 我们正式的定義嵌套的關係如下:

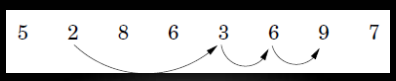
  盒子D=(d_1, d_2, ..., d_n)和E=(e_1, e_2, ..., e_n)，當存在一種1...n的排列p使得($d_{p(1)}$, $d_{p(2)}$, ..., $d_{p(n)}$)能夠匹配(e_1, e_2, ..., e_n)，即對於所有的$1 \le i \le n$都有$d_{p(i)} < e_i$，那麼就認為D可以嵌入E。

Methods 1: DAG (最長路徑問題)

- 先將盒子D=(d1, d2, ..., dn)和E=(e1, e2, ..., en)分別依小排到大排序好。
- 當存在一種1...n的排列p使得($d_{p(1)}$, $d_{p(2)}$, ..., $d_{p(n)}$)能夠匹配於盒子E=(e_1, e_2, ..., e_n)。即對於所有的$1 \le i \le n$都有$d_{p(i)} < e_i$，那麼就認為D可以嵌入E。
- 故我們假設若a"可嵌套於"b，那麼就是G[a][b]=1。然後這個問題就可以轉化成求一個最長路徑長度的大小。

Methods 2: LIS (最長遞增子序列)

- 給定一個序列$a_1, a_2, \dots, a_n$
- 定義原序列中的子序列為原序列元素的一個子集，且子序列各元素的相對應順序不變$a_{i_1}, a_{i_2}, \dots, a_{i_n}$，且$1 \le i_1 < i_2 < \cdots i_k \le n$。由此找出給定序列的最長遞增子序列。
- 例: 序列5,2,8,6,3,6,9,7的最長遞增子序列是2,3,6,9

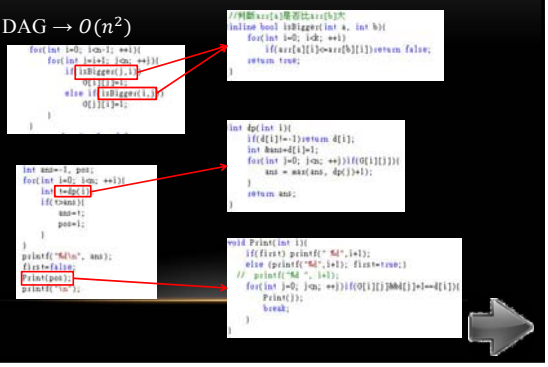


Sample for factory

- 工廠貨品範例
- 家中空箱
- 孩子學習
- 物流士

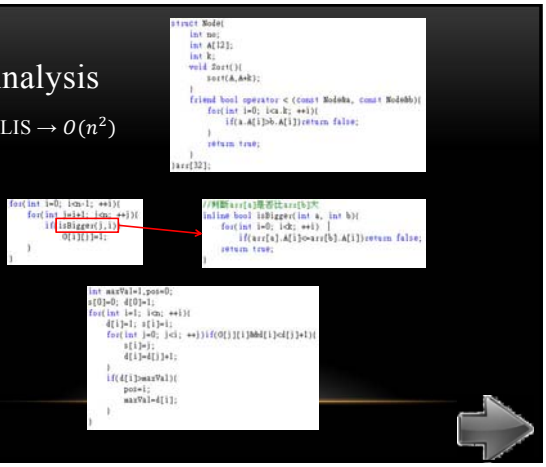
Analysis

- DAG → $O(n^2)$



Analysis

- LIS → $O(n^2)$



分工方式

姓名	內容
林明慶	Video, Coding, PowerPoint, Analysis
林暐智	Collecting data of DAG
莊士鋒	Collecting data of LIS


~Thanks for your Attention~