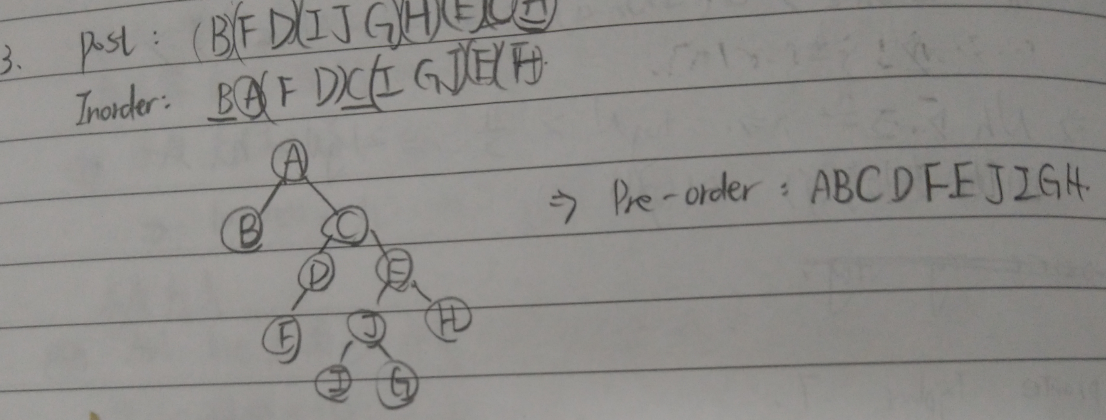
1. 名詞說明
2. deep learing : 機器學習的分支，是一種以人工神經網路為架構，希望把資料透過多個處理層(layer)中的線性或非線性轉換(linear or non-linear transform)，對資料進行表徵學習的演算法
3. block chain : 是藉由密碼學串接並保護內容的串連文字記錄（又稱區塊）；每一個區塊包含了前一個區塊的加密雜湊、相應時間戳記以及交易資料，這樣的設計使得區塊內容具有難以篡改的特性
4. p2p network : 是無中心伺服器、依靠使用者群（peers）交換資訊的網際網路體系，對等網路的每個使用者端既是一個節點，也有伺服器的功能，它的作用在於，減低以往網路傳輸中的節點，以降低資料遺失的風險
5. gpu computing : 是利用處理圖形任務的[圖形處理器](https://zh.wikipedia.org/wiki/%E5%9C%96%E5%BD%A2%E8%99%95%E7%90%86%E5%99%A8)來計算原本由[中央處理器](https://zh.wikipedia.org/wiki/%E4%B8%AD%E5%A4%AE%E5%A4%84%E7%90%86%E5%99%A8)處理的通用計算任務，由於現代圖形處理器有強大的並列處理能力和可程式化管線，令圖形處理器也可以處理非圖形資料
6. 搜尋演算法
7. 比較semantic search & social search :

semantic search → 根據用戶所輸入的內容，除了考慮輸入的文字還考慮其上下文語意情境作為分析的資訊

social search → 在一般搜尋結果中，整合來自社群的推薦與評價內容，讓用戶可以在看到搜尋引擎的判斷之外，也同時接收自己社群好友的喜好

1. tree traversal



1. heap adjust

void build\_heapify\_r(int arr[], int root, int n) {

int j = root \* 2 + 1;

if (j < n) {

if (j + 1 < n) {

if (arr[j] < arr[j + 1])

j = j + 1;

}

if (arr[j] > arr[root]) {

swap(arr[j], arr[root]);

build\_heapify\_r(arr, j, n);

}

}

}

void swap(int arr[], int curr, int n){

int temp = arr[curr];

arr[curr] = arr[n];

arr[n] = temp;

}

1. deadlock prevent & avoidance差別

→ 針對會造成deadlock的必要四條件做預防，不讓其發生，avoidance則是在process提出資源申請時，執行banker' algo判斷是否為safe state，若是則分配，反之

1. os排班 - average trunaround time

回覆時間 (turnaround time)：針對單一行程來進行觀察時，要知道其執行開始到結束總共花多少時間，包括等待載入記憶體的時間、在就緒佇列等待時間、在CPU執行的時間、和處理I/O所花的時間，通常回覆時間受限於輸出裝置的速度

等待時間 (Waiting Time) : Process待在Ready Queue等待獲取CPU的時間總和

Turnaround time = Exit time - Arrival time || Turnaround time = Burst time + Waiting time

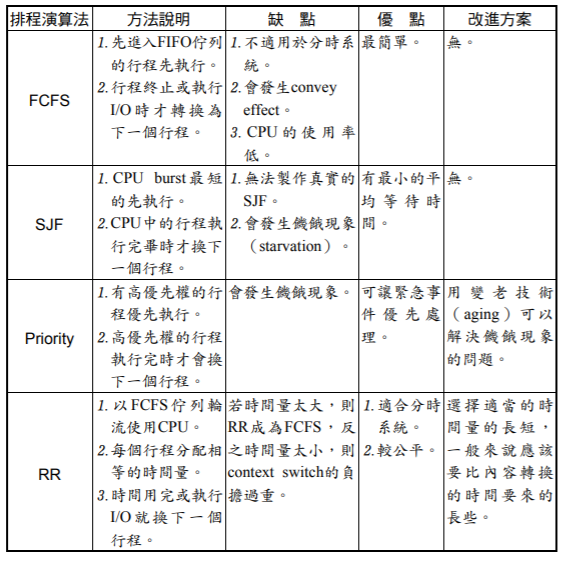
Waiting time = Turnaround time - Burst time

1. FCFS : 10.5
2. Nopreemptive shortest-job-fisrt : 9

→ 先到先執行不能打斷，若執行間有大於兩個process等待的話，依照誰的執行時間短誰先執行

→ https://www.guru99.com/shortest-job-first-sjf-scheduling.html

1. preemptive shortest-job-fisrt : 8.75



1. late binding
2. 指在程式執行時根據傳入函式的物件或是型別，決定執行的起始位置也就是呼叫哪一個函式的機制
3. 多型 : 指為不同[資料類型](https://zh.wikipedia.org/wiki/%E6%95%B8%E6%93%9A%E9%A1%9E%E5%9E%8B)的實體提供統一的[介面](https://zh.wikipedia.org/wiki/%E4%BB%8B%E9%9D%A2_(%E7%A8%8B%E5%BC%8F%E8%A8%AD%E8%A8%88))並根據傳入的資料類型做出對應的動作，由於傳入的物件是在執行期間才決定，因此需要late binding的機制有關

[補充] 繫結 (binding) 機制是指 編譯器/執行環境 針對程式裡某一函式呼叫敘述，如何決定呼叫哪一個函式的機制

1. divide and conquer & merge sort
2. Devide: 將原問題分成若干個仔問題

Conquer: 遞迴解決各個子問題；當子問題夠小時則直接解

Combine: 將子問題的解合併成原問題的解

