HW1

1.請設計一個具運算子優先序能力的計算機(先算乘除後算加減)

2.至少能運算加、減、乘、除、次方及小括號

(可增加其他運算子或函數)

3.顯示過程中所使用堆疊中的元素

4.輸出運算式後序運算式

5.輸出運算式的前序運算式(加分項)

6.輸出計算結果

HW2

1.由檔案(c:\data.txt)中讀入以level order為序的binary tree : 無節點處以 @@ 表示,每列資料字串為tree 同一個level上的node, tree 不一定是complete binary tree, 同一level列尾部的空節點不會記錄在檔案內。請先以陣列讀入上述檔案中的資料,再以指標方式設計tree的資料結構。請輸出此二元樹的前序(preorder)、中序(inorder)、後序(postorder)、階序(level order)追蹤

例如:輸入檔內容

A

B C

D @@ E

@@@@@@ F

2.若能顯示tree、拜訪的歷程者加分

EX1

1.請以動態陣列實作一元多項式,並盡量節省記憶體使用.

2.由檔案中讀入2多項式的各項非0係數及次方,

例如:輸入2 2 3 -3 0表示多項式有2項為2X^3-3

3.輸出多項式之和(80分)及積(20分):僅輸出各項非0係數及次方

例如:

2 2 3 -3 0

2 1 1 1 0

輸出

和:2 3 1 1 -2 0

積:2 4 2 3 -3 1 -3 0

EX2

自檔案data.txt讀入無向圖:先讀入圖的頂點數n及邊數e,再讀入所有邊v,u。頂點為編號1~n。輸出所有圖中的articulation point後再逐一輸出圖中的雙向連通元件。輸入及輸出範例請至教材目錄中下載。

EX3

1.以(\*lchild,data,\*rchid)格式,建置可以做一般搜尋及rank搜尋的BST(自工作目錄的data.txt讀入指令)

2.自檔案讀入操作節點插入、刪除、查詢指令,更新BST並輸出結果。

2.1插入二元搜尋樹節點,從檔案讀入 I 值:若成功插入節點則輸出節點所在level及其order,否則輸出"節點重複"

2.2刪除節點,從檔案讀入 D 值:若刪除成功,輸出節點所在level及其order,否則輸出"無此節點"

2.3搜尋第n大節點,從檔案讀入 R n:輸出節點值、所在level及其order,若找不到則輸出"R無此節點"

2.4搜尋節點,從檔案讀入 S 值:輸出所在level及其order,若找不到則輸出"S無此節點"

3.輸出postorder travsal sequences,從檔案讀入 P 0

4.輸出preorder travsal sequences,從檔案讀入 P 1

5.輸出inorder travsal sequences,從檔案讀入 T 0

6.從檔案輸入 @ 時結束程式:需回收樹節點

EX4

1.自檔案讀入m,n,s,e表示有向量G有n個頂點,m個有向邊,s為出發頂點,e為終點.

2.讀取m個有向邊及其長度(日數),該邊代表一個專案中的工作名稱為aij,<i,j>為有向邊;若存在axi的工作,表示aij需待所有axi完成後才能開始進行。

3.請找出最早完成所有工作的日數(70%)

4.請找出專題中的critical path(30%)

EX5

1.請分別實作Insertion Sort/Quick Sort/Merge Sort/Heap Sort/Radix Sort並以實驗法比較其平均執行效能。

2.實驗資料集大小分別為500,5000,10000,20000~90000筆,每個資料集各500組進行實驗以取得平均執行時間,輸出成csv後以Excel產生折線圖比較之。

3.資料集中資料可設計成結構紀錄,其中排序的鍵值為浮點數以亂數產生,資料集中其他非鍵值資料可以使用固定值或亂數產生(若僅只有單純使用鍵值陣列實作亦可得基本分)。【請注意】不同演算法實驗中需使用相同的資料集(例如產生一組500筆資料,在進行排序前複製陣列至供演算法排序陣列,演算法在複製後陣列中操作,不會變更原始資料集)

4.統計各資料集的資料比較次數及交換次數,輸出不同大小資料集的平均及最大、最小次數(加分項)。

5.merge sort若能結合List sort以減少資料交換次數可加分。

6.請繳程式碼及一頁A4文件簡要說明你採用的資料結構及演算法(各演算法有不同版本)