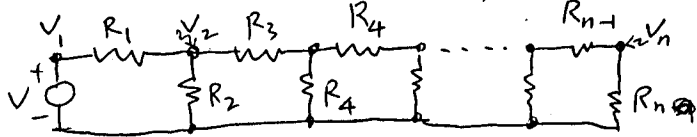


1. 下面為電路學上的線路, 請



- (a) 設計一資料結構, 用以表示上面的線路。 5%
- (b) 該結構如何用現有的程式語言表示。 5%
- (c) 根據 (a) 之結果, 如何判斷二種線路, 是否相同 5%
- (d) 有一個 50 英的線路, (英, 如上面的 V_1, V_2, \dots, V_n) 請將 (a) 之結構所使用的記憶體, 其使用率提高。 5%
- (e) 如果使用率不高, 如何改善。 5%
2. (a) 何謂 Tournament sort, 請舉例說明。 10%
- (b) 何謂 Heapsort, 亦請舉例說明。 10%
- (c) 兩者之間有何關係 5%
3. (a) 請將下列 (A), (B) 兩組名詞間, 有關係者用線連結起來。 10%

(A)

- ① Bucket
- ② list
- ③ STACK
- ④ Hash function
- ⑤ insertion sort
- ⑥ Symbol table

(B)

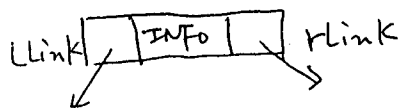
- ① Hash addressing technique
- ② shell sort
- ③ recursive program
- ④ Hanoi problem
- ⑤ Dynamic storage Allocation.

(b) 請用 Binary Tree 表示 $A = A + \sin(x + B(\pi))$ 5%

4. ① 何謂 Internal Search. 3%
- ② 何謂 External Search 3%
- ③ 以上兩種之考慮之重點 ~~有何不同~~ 有何不同? 3%
- ④ 請舉一例說明可用於 Internal Search ~~的~~ 方法, 但 ~~不~~ 一定可用於 External Search 的方法. 4%
- ⑤ 何謂 associative Memory? 3%
- ⑥ 用何種 addressing technique, 可使 external data access 符合 ⑤ 之定義, 為何? 4%

5. 下面 (A), (B) 兩個 procedure 係用 pascal 寫成, 功用相同, 將 Tree "root" 內的資料印出來,

- (a) 請問將 Tree 印出的次序為何? 5%
(該 Tree 為 Binary tree, 其資料結構為



- (b) 將 (B) 程式內容空白處填入 pascal 的指述 (每一個空白, 填一個指述), 使 (A), (B) 兩程式功能相同。 10%

```

TYPE  NODE = Record
        lLink: int ↑ node;
        info: int;
        rLink: ↑ node;
    END;
    
```

```

(A)  PROC  visittree (root: ↑ node)
    BEGIN
        IF root = nil THEN return
        ELSE BEGIN
            visittree (root.lLink);
            write (root.info);
            visittree (root.rLink);
        END;
    END;
    
```

```

(B) PROC visitree (root: ↑node);
    BEGIN
        IF root = NIL THEN RETURN
        ELSE
            BEGIN
                PUSH (root);
                root := root↑.llink;
            1: WHILE root ≠ NIL DO
                BEGIN
                    (1) ;
                    (2) ;
                END;
                POP (root);
                WHILE root ≠ EOS DO
                    (* EOS means END OF STACK *)
                    BEGIN
                        WRITE (root↑.info);
                        IF root↑.rlink ≠ NIL THEN
                            BEGIN
                                (3) ;
                                (4) ;
                            END
                        ELSE (5) ;
                    END;
                END;
            END;
        END;
    END;

```

(註: PUSH 及 POP 為 STACK 的兩個標準動作)

~~~~~END~~~~~