96 成功大學　程式設計

一、Algorithm

1. TRUE (用 lim 法去解)
2. $T(n) = \theta(n^2\sqrt{n})$

3.

(a) 在有向圖之 subgraph 滿足，

(1) subgraph 中任兩點均兩兩有 path

(2) 若再加入一個 vertex 則無(1)之特性

(b)

Step1: 對 G 作 DFS(G), 記下 finish time

Step2: 作 G^T (即(a, b) -> (b, a))

Step3: 作 DFS(G^T) 且依照 Step1 之 finish time 之遞減順序，則其在 steps 所求之 forest 即為 SCC

4.

(a) 利用 Greedy 之策略，針對每點進行檢查，算出 shortest path

(b) 主要 Algorithm 如下

```
for( i=1 to n )
  for( j=1 to n )
      for( k=1 to n )
      {
          C[ i, j ] <- min{ C^(k-1)[ i, j ], C^(k-1)[ i, k ] + C^(k-1)[ k, j ] }
      }
```

Time Complexity : $O(n^3)$

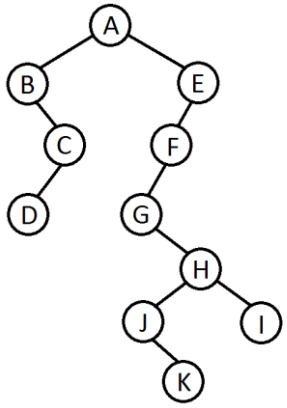
二、Data Structure

1. 1001*1001 = 1002001
2. T T F F F
   T T F F F

3.

(1) BDCGJKHIFEA

(2)



(3) ABCDEFGHJKI

(4) BDCAGJKHIFEA

(5) 原諒我用小畫家很難畫= =

4.

(a) Fig.1 (b) Fig.2 (c) Fig.6 (d) Fig.3 (e) Fig.4

```
void adjust(element list[], int root, int n)
{
    int child, rootkey;     element temp;
    temp=list[root];        rootkey=list[root].key;
    child=2*root;
    while (child <= n) {
        if ((child < n) && (list[child].key < list[child+1].key)) child++;
        if (rootkey > list[child].key) break;
        else {
            list[child/2] = list[child];
            child *= 2;
        }
    }
    list[child/2] = temp;
}

void heapsort(element list[], int n)
{
    int i, j;
    element temp;
    for (i=n/2; i>0; i--) adjust(list, i, n);
    for (i=n-1; i>0; i--) {
        SWAP(list[1], list[i+1], temp);
        adjust(list, 1, i);
    }
```

```
}
```