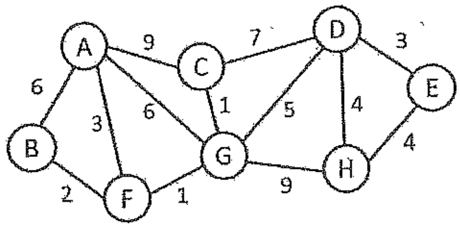
國立交通大學 107 學年度碩士班考試入學試題

科目：資料結構與演算法(1101)　　考試日期:107 年 2 月 2 日 第 1 節

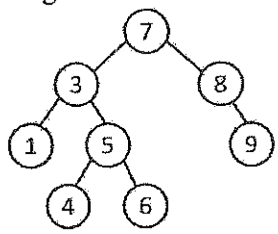
系所班別：資訊聯招　　

【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

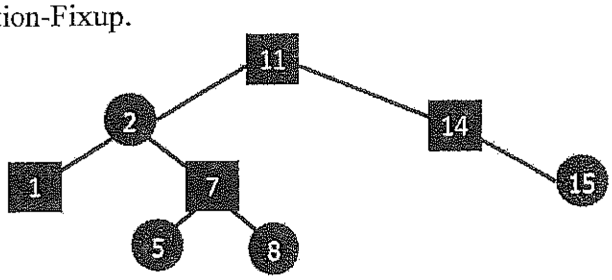
1. (5%) What is the summed edge weight if you transform the following graph to a minimum spanning tree?



2. (10%) What are the sequences if the following tree is traversed by in-order and post-order, respectively?



3. (10%) The figure below shows an RB-tree. The black and the red nodes are visualized in squares and circles, respectively. What will be the tree look like once a new element {9} is added into the tree? Please draw the black node using a square and the red node using a circle in your answer. Note that you have to draw all the steps in Insertion-Fixup.



4. (5%) Consider the following function. Please derive the time complexity of function foo using recurrence relation. Please show your derivation step by step.

```
int i[n]; /* Assume that the values of all integers in i[n] have been initialized */
int foo(int a, int b, int c) {
  int d;
  if (a>b) return -1;
  d=(a+b)/2;
  if (c==i[d])
      return d;
  else if (c>i[d])
      return foo(d+1,b,c);
  else
      return foo(a,d-1,c);
}
```

5. (10%) Please show the result of the following code.

```
#include <iostream>
using namespace std;

struct node {
  int value;
  struct node * left, * right;
} n1, n2, n3, n4, n5, n6, n7;

int flag=0, sum=0;

void foo1(struct node * p) {
  struct node * t;
  if (p!=NULL) {
      t=p->left;
      p->left=p->right;
      p->right=t;
      foo1(p->left);
      foo1(p->right);
  }
}

void foo2(struct node * p) {
  if (p!=NULL) {
      foo2(p->left);
      foo2(p->right);
      if (flag%2==0) sum+=p->value;
  }
  flag++;
}

int main() {
  n1.value=6; n2.value=10;
  n3.value=1; n4.value=8;
  n5.value=12; n6.value=14;
  n7.value=4;
  n1.left=&n2; n1.right=&n3;
  n2.left=&n4; n2.right=&n5;
  n3.left=&n6; n3.right=&n7;
  n4.left=n4.right=NULL;
  n5.left=n5.right=NULL;
  n6.left=n6.right=NULL;
  n7.left=n7.right=NULL;
  foo1(&n1);
  foo2(&n1);
  cout<<sum;
}
```

【不可使用計算機】*作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

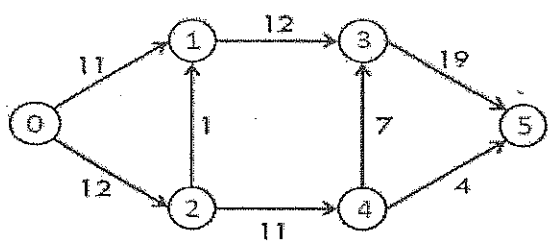
6. (10%) Please show the result of the following code.

```
#include <iostream>
using namespace std;

struct node {
  int value;
  struct node * next;
} n1, n2, n3, n4, n5, * f;

void foo1(struct node * p) {
  for(;p!=NULL && p->next!=NULL;p=p->next)
      if (p->value>p->next->value) {
          int t;
          t=p->value;
          p->value=p->next->value;
          p->next->value=t;
      }
}

void foo2(struct node * p) {
  struct node *q, * r;
  if (p==NULL) return;
  q=p->next; p->next=NULL;
  for(;q!=NULL;p=q,q=r) {
      r=q->next; q->next=p;
  }
  f=p;
}

int bar(struct node *p) {
  int flag, sum;
  for(flag=0,sum=0;p!=NULL;p=p->next,flag++)
      if (flag&1==1)
          sum+=p->value;
  return sum;
}

void bar2(struct node * p) {
  for(;p!=NULL;p=p->next)
      cout<<p->value<<" ";
}

int main() {
  n1.value=6; n2.value=3; n3.value=8;
  n4.value=10;    n5.value=1;
  n1.next=&n2;    n2.next=&n3;    n3.next=&n4;
  n4.next=&n5;    n5.next=NULL;
  f=&n1;
  foo1(f);
  foo2(f);
  cout << bar(f);
}
```

7. (15%) Let G be a directed graph and w be the edge weights of G.

```
Mystery-Algorithm(G, w)
    1.
    create G' where G'.V = G.V ∪ {s},
        G'.E = G.E ∪ {(s, u): for u in G.V}, and
        w(s, u) = 0 for u in G.V
    2.
    if Bellman-Ford(s) == False
        return "The input graph has a negative weight cycle"
    else:
        for each vertex v in G'.V:
            h(v) = distance(s, v) computed by the Bellman-Ford
        for each edge (u, v) in G'.E:
            w'(u, v) = w(u, v) + h(u) - h(v)
    3.
        D = new matrix of distances initialized to infinity
        for each vertex u in G.V:
            run Dijkstra(G, w', u) to compute distance'(u, v)
                              for all v in G.V
            for each vertex v in G.V:
                D(u, v) = distance'(u, v) + h(v) - h(u)
      return D
```

(a) (5%) What are the function of the above algorithm and its complexity with a Fibonacci heap?
(b) (10%) Why Dijkstra algorithm can be used in Step 3? Explain your reason.



8. (10%) Consider the above flow network, where the integer on each arc indicates its capacity and the numbers in the circles are labels of the nodes. Node 0 is the source and node 5 is the sink.
(a) (5%) Use Edmonds-Karp algorithm to find the maximum flow. Illustrate your answer step by step for the first 5 iterations. You should show the residual networks and the corresponding flows starting with the initial flow 0.
(b) (5%) Find a minimum cut.

9. (10%) We define the maximum subarray of an array A to be the nonempty, contiguous subarray of A whose values have the largest sum. Fill in the blank (a), (b) in the following C++ function so that it returns the sum of the maximum subarray of the input array A in O(n) time. Note that A is a length-n array whose values are placed in A[0], A[1], . . . , A[n − 1].

```
int maxSubarray(int A[], int n){

  for(int i=1; i<n; ++i){
    A[i] += A[i-1];
  }
  int ans = A[0];
  int k = 0;

  for(int i=0; i<n; ++i){
    ans = max(ans, ___(a)___);
    k = min(k, ___(b)___);
  }

  return ans;
}
```

10. (15%) A Hamiltonian path of a graph G is a path that visits each node in G exactly once. Suppose that there is an $O(n^7)$-time algorithm that decides HamP(G) for any n-node graph G.

HamP(G)
Input: an undirected graph G
Output: "Yes", if G has a Hamiltonian path; "No", otherwise.

Give an $O(n^7)$-time algorithm that decides HamEx(G, x) for any n-node graph G, and prove the correctness of your algorithm. Note that your algorithm must have running time $O(n^7)$. No partial credit will be given if your algorithm runs asymptotically slower.

HamEx(G, x)
Input: an undirected graph G, and a node x in G
Output: "Yes", if G has a Hamiltonian path from node u to node v so that $u \neq x$ and $v \neq x$; "No", otherwise.