浙江大学 2017-18 秋冬《数据结构基础》期末模拟练习

开始时间 1/1/2016, 12:00:00 AM
结束时间 1/18/2038, 12:00:00 AM
答题时长 120 分钟
考生王子腾
得分 94
总分 100

判断题总分：20 得分：16

1-1

If the most commonly used operations are to visit a random position and to insert and delete the last element in a linear list, then sequential storage works the fastest. (2 分)

◉ T ○ F

评测结果：答案正确（2 分）

1-2

Given a binary search tree with 20 integer keys which include 7, 8, and 9, if 7 and 9 are on the same level, then 8 must be their parent. (2 分)

○ T ◉ F

评测结果：答案正确（2 分）

1-3

In a directed graph, the sum of the in-degrees must be equal to the sum of the out-degrees of all the vertices. (2 分)

◉ T ○ F

评测结果：答案正确（2 分）

1-4

The minimum spanning tree of a connected weighted graph with vertex set V={ A, B, C, D, E } and weight set W={ 1, 3, 2, 5, 1, 7, 9, 8, 4} must be unique. (2 分)

○ T ◉ F

评测结果：答案错误（0 分）

1-5

For a sequentially stored linear list of length N, the time complexities for deleting the first element and inserting the last element are O(1) and O(N), respectively. (2 分)

○ T ◉ F

评测结果：答案正确（2 分）

1-6

If the preorder and the postorder traversal sequences of a binary tree have exactly the opposite orders, then none of the nodes in the tree has two subtrees. (2 分)

◉ T ○ F

评测结果：答案正确（2 分）

1-7

If there are less than 20 inversions in an integer array, then Insertion Sort will be the best method among Quick Sort, Heap Sort and Insertion Sort. (2 分)

◉ T ○ F

评测结果：答案正确（2 分）

1-8

O(N2) is the same as O(1+2+3+…+N). (2 分)

○ T ◉ F

评测结果：答案错误（0 分）

1-9

If 7 elements have been stored in a hash table of size 13 at positions { 0, 1, 3, 4, 9, 10, 12 }, and the hash function is H(x)=x%13. Then an empty spot can't be found when inserting the element 26 with quadratic probing. (2 分)

◉ T ○ F

评测结果：答案正确（2 分）

1-10

After the first run of Insertion Sort, it is possible that no element is placed in its final position. (2 分)

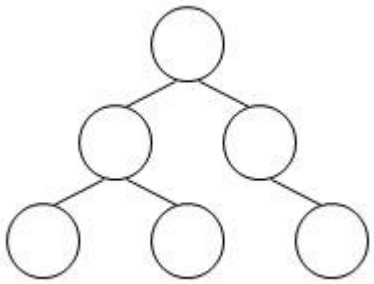
◉ T ○ F

评测结果：答案正确（2 分）

单选题总分：60 得分：60

2-1

Given the structure of a binary search tree (as shown in the figure), which one of the following insertion sequences is impossible? (3 分)

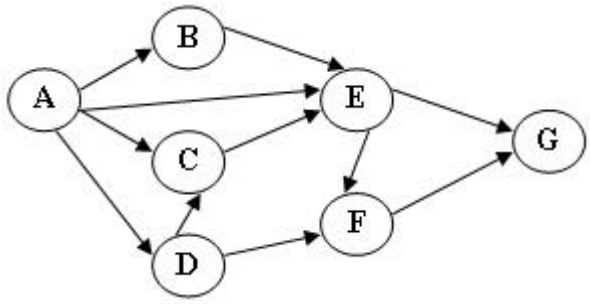


1. ○ `81 68 92 96 19 74`
2. ○ `81 68 74 92 19 96`
3. ◉ `81 92 74 68 19 96`
4. ○ `81 92 96 68 74 19`

评测结果：答案正确（3 分）

2-2

The figure shows an AOV network. Which one of the following is a possible topological order of the network? (3 分)

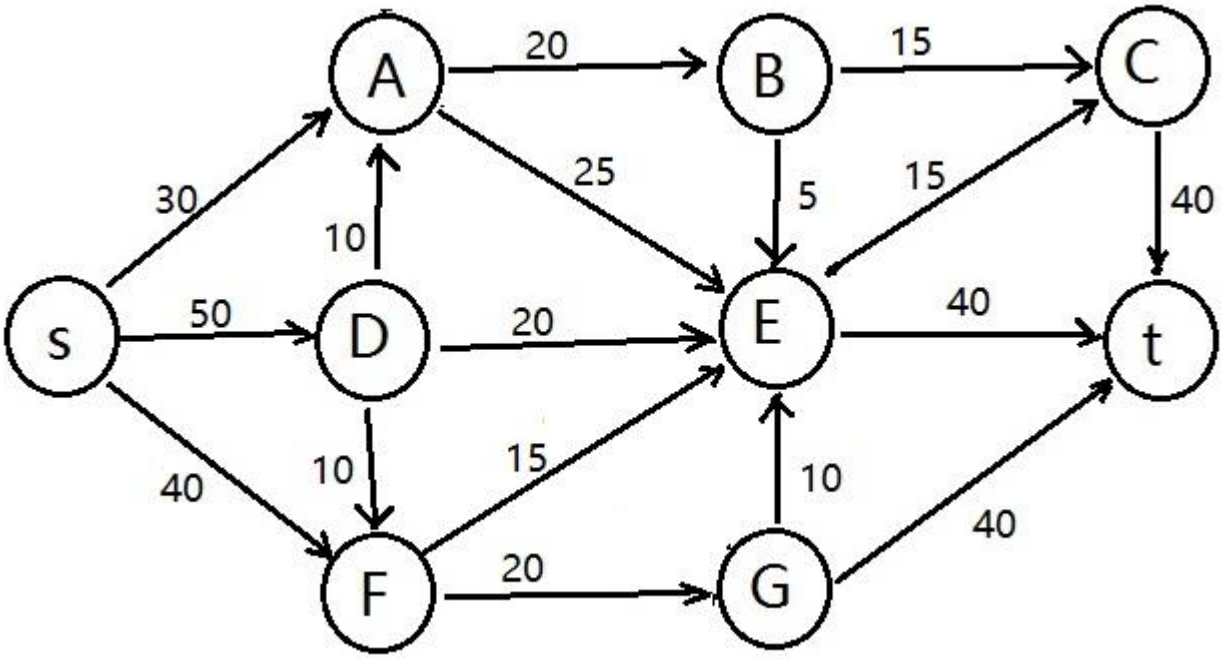


1. ABCDFEG
2. ADFCEBG
3. ACDFBEG
4. ABDCEFG

评测结果：答案正确（3 分）

2-3

The maximum flow in the following network is: (3 分)



1. 100

2. ○ 95
3. ◉ 90
4. ○ 85

评测结果：答案正确（3 分）

2-4

To sort { 8, 3, 9, 11, 2, 1, 4, 7, 5, 10, 6 } by Shell Sort, if we obtain (4, 2, 1, 8, 3, 5, 10, 6, 9, 11, 7) after the first run, and (1, 2, 3, 5, 4, 6, 7, 8, 9, 11, 10) after the second run, then the increments of these two runs must be __ , respectively. (3 分)

1. ○ 3 and 1
2. ◉ 3 and 2
3. ○ 5 and 2
4. ○ 5 and 3

评测结果：答案正确（3 分）

2-5

If the hash values of n keys are all the same, and linear probing is used to resolve collisions, then the minimum total number of probings must be __ to insert these n keys . (3 分)

1. ○ n(n−1)/2
2. ○ 2n+1
3. ○ 2(n+1)
4. ◉ n(n+1)/2

评测结果：答案正确（3 分）

2-6

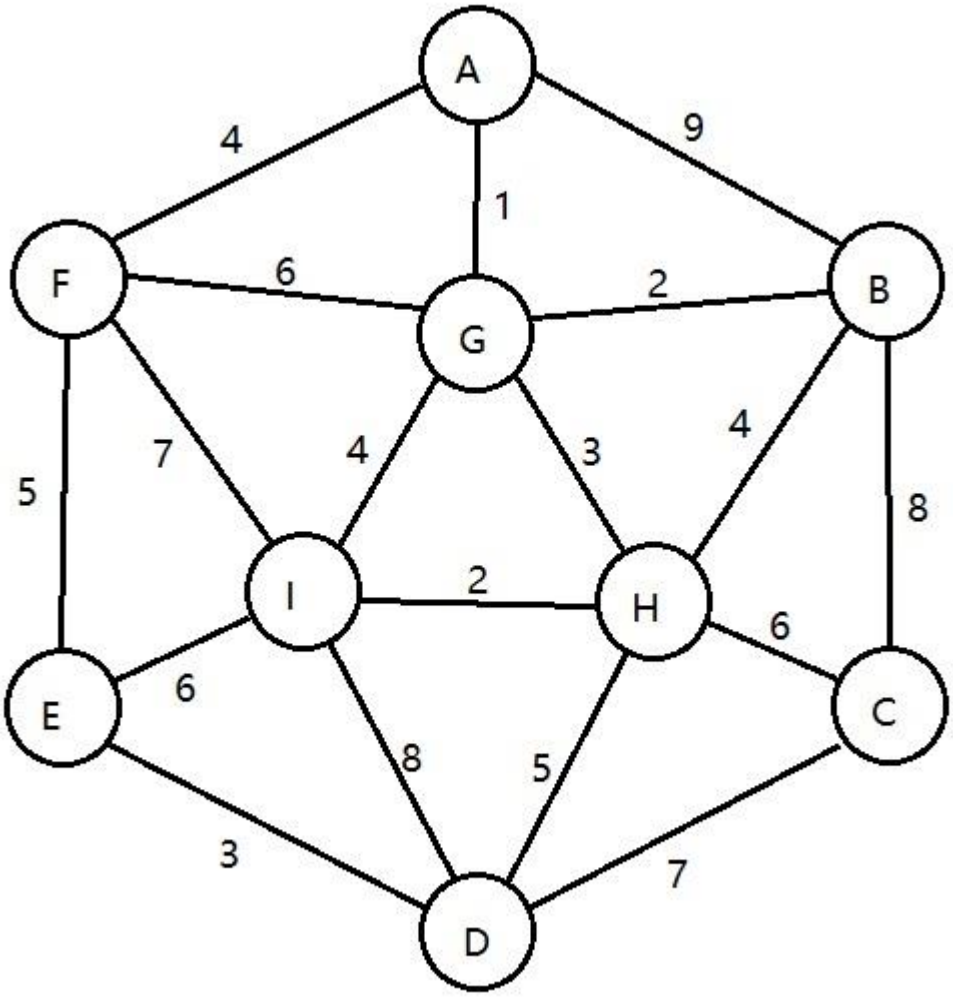
Given a tree of degree 6. Suppose that the numbers of nodes of degrees 1, 2, 3, 4, 5, 6 are 3, 5, 2, 3, 1, 2, respectively. Then the number of leaf nodes must be: (3 分)

1. ○ 29
2. ○ 31
3. ◉ 33
4. ○ 35

评测结果：答案正确（3 分）

2-7

Given a weighted graph as shown by the figure. Which one of the following statements is TRUE about its minimum spanning tree? (3 分)



1. ◉ The minimum spanning tree is not unique and the total weight is 26.

2. ○ The minimum spanning tree is not unique and the total weight is 28.
3. ○ If the weight of edge (E,F) was 3, then the minimum spanning tree would not be unique.
4. ○ If the weight of edge (D,E) was 9, then the minimum spanning tree would not be unique.

评测结果：答案正确（3 分）

2-8

The inorder and the postorder traversal sequences of a binary tree are `1 2 3 4 5 6 7` and `2 3 1 5 7 6 4`, respectively. Then the preorder traversal sequences is: (3 分)

1. ○ `4 2 1 3 6 5 7`
2. ○ `4 1 2 3 6 7 5`
3. ◉ `4 1 3 2 6 5 7`
4. ○ `4 3 1 2 6 7 5`

评测结果：答案正确（3 分）

2-9

Suppose that the level-order traversal sequence of a max-heap is {98, 72, 86, 60, 65, 12, 23, 50}. Use the linear algorithm to adjust this max-heap into a min-heap, and then run DeleteMin twice. The inorder traversal sequence of the resulting tree is: (3 分)

1. ○ 50, 60, 98, 65, 72, 86
2. ◉ 98, 60, 65, 50, 86, 72
3. ○ 98, 60, 50, 65, 86, 72
4. ○ 72, 60, 65, 50, 98, 86

评测结果：答案正确（3 分）

2-10

Given an undirected graph, and the edge set of a DFS from V0 as: { (V0,V1) , (V0,V4) , (V1,V2) , (V1,V3) , (V4,V5) , (V5,V6) }. Which one the following **cannot** be the sequence of another DFS? (3 分)

1. ○ V0, V1, V2, V3, V4, V5, V6
2. ○ V0, V3, V1, V2, V4, V5, V6
3. ◉ V0, V1, V4, V5, V6, V2, V3
4. ○ V0, V5, V4, V6, V1, V2, V3

评测结果：答案正确（3 分）

2-11

Rehashing is required when an insertion to a hash table fails. Which of the following is **NOT** necessary to rehashing? (3 分)

1. ◉ change the collision resolution strategy
2. ○ use a new function to hash all the elements into the new table
3. ○ build another table that is bigger than the original one
4. ○ scan down the entire original hash table for non-deleted elements

评测结果：答案正确（3 分）

2-12

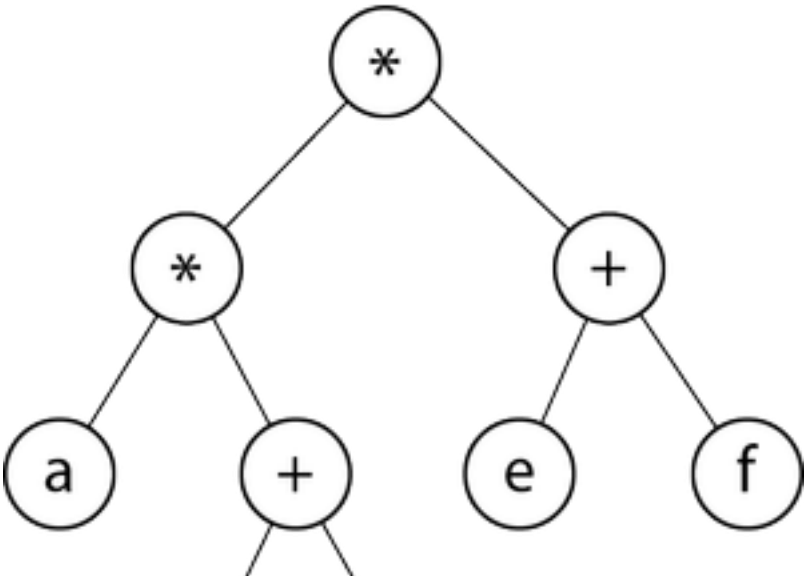
For a directed graph, comparing to the representation of the adjacency lists, the adjacency matrix representation is easier to: (3 分)

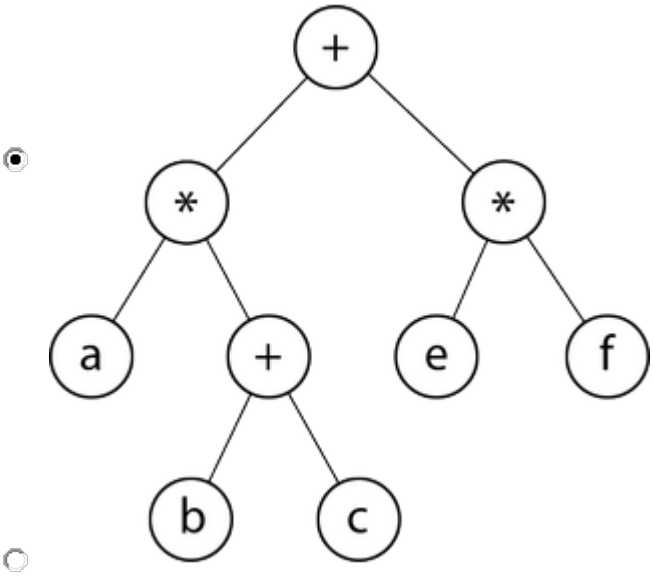
1. ◉ find the in-degree of a vertex
2. ○ find the vertices that are adjacent from a vertex
3. ○ do DFS
4. ○ do BFS

评测结果：答案正确（3 分）

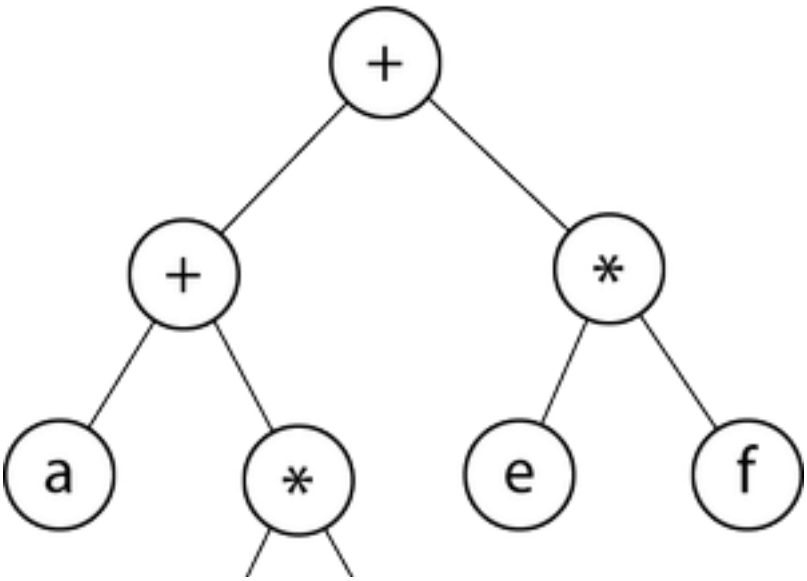
2-13

Which one of the following is the expression tree corresponding to the postfix expression `abc+*ef+*`? (3 分)

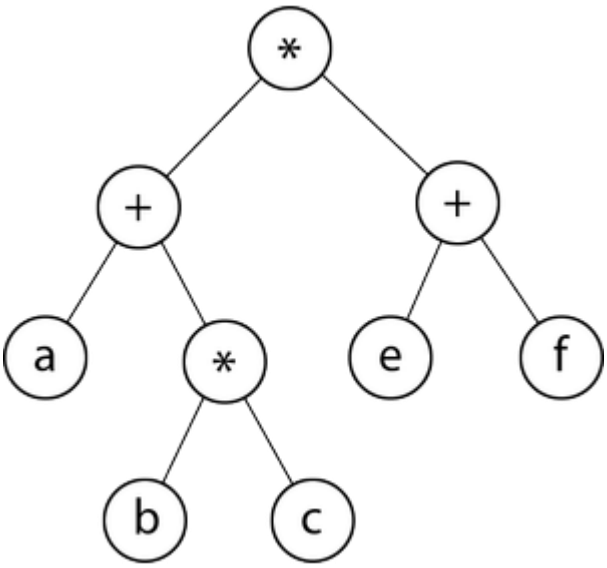


1. 

2.



3.



4.

评测结果：答案正确（3 分）

2-14

For an in-order threaded binary tree, if the pre-order and in-order traversal sequences are `B E A C F D` and `A E C B D F` respectively, which pair of nodes' left links are both threads? (3 分)

1. B and E
2. E and F
3. C and F
4. C and D

2. ○ E and F
3. ○ C and F
4. ◉ C and D

评测结果：答案正确（3 分）

2-15

Suppose that an array of size `m` is used to store a circular queue. If the head pointer `front` and the current size variable `size` are used to represent the range of the queue instead of `front` and `rear`, then the maximum capacity of this queue can be: (3 分)

1. ○ `m-1`
2. ◉ `m`
3. ○ `m+1`
4. ○ cannot be determined

评测结果：答案正确（3 分）

2-16

A relation R is defined on a set S. If for any elements a and b in S, we have that if "a R b" is true, then "b R a" must be true, then R is said to be __ over S. (3 分)

1. ○ reflexive
2. ○ transitive
3. ◉ symmetric
4. ○ invertive

评测结果：答案正确（3 分）

2-17

Given that the pushing sequence of a stack is { 1, 2, ···, n } and popping sequence is { p1, p2, ···, pn }. If p2=n, how many different possible popping sequences can we obtain? (3 分)

1. ○ 2
2. ○ 1
3. ○ n
4. ◉ n−1

评测结果：答案正确（3 分）

2-18

Quick Sort can be implemented by a recursive function `void Qsort( ElementType A[ ], int Left, int Right )`. If we are to implement the function `Qsort()` in a **non-recursive** way with a stack, which of the following should be packed as the elements of the stack? (3 分)

1. ○ value of pivot
2. ○ index of pivot
3. ◉ `Left` and `Right`
4. ○ only `Left` or `Right`

评测结果：答案正确（3 分）

2-19

When inserting a new key `a` into a binary search tree `T` with 1022 nodes, the worst-case number of comparisons between `a` and the keys already in `T` is in the range of: (3 分)

1. ○ [11, 1022]
2. ◉ [10, 1022]
3. ○ [10, 1023]
4. ○ [11, 1023]

评测结果：答案正确（3 分）

2-20

Which one of the following is a possible postorder traversal sequence of a binary search tree? (3 分)

1. ○ `3 1 6 2 5 8 9 7 4`
2. ○ `3 1 2 6 5 8 9 7 4`
3. ◉ `1 3 2 6 5 8 9 7 4`
4. ○ `1 3 2 6 8 5 9 7 4`

评测结果：答案正确（3 分）

程序填空题总分：12 得分：10

5-1

The function is to turn an array `A[]` of `N` elements into a max-heap.

```
#define leftchild(i) ( 2*(i)+1 )

void BuildMaxHeap( ElementType A[], int N )
{  int i, j, child;
   ElementType Tmp;

   for ( i = (N-1)/2; i >= 0; i-- ) {
      j = i;
      for ( Tmp = A[j]; leftchild(j) < N; j = child ) {
         child = leftchild(j);
         if ( (child+1<N)&&(A[child+1]>A[child]) (2 分))
            child ++;
         if ( A[j]<A[child] (2 分))   A[j] = A[child];
         else  break;
      }
      A[j]=Tmp (2 分);
   }
}
```

Thanks to DOU Yan from Yanshan University for the correction!

评测结果：部分正确（4 分）

序号	结果	得分
0	答案正确	2
1	答案错误	0
2	答案正确	2

5-2

Given an array `a[]`of `n` integers, the function `MissingMax` is to find and return the maximum negative integer which is **NOT** in the array. For

example, given { -3, -1, 8, 1, 0 }, -2 is the largest negative integer which is missing.

```
int MissingMax( int a[], int n )
{
    int i, j, max, missing = -1;

    for( i=0; i<n; i++ ){
        max = i;
        for( j = i+1; j < n; j++ )
            if ( a[j]>a[max] (3 分))  max = j;
        if ( max != i )  swap(a[i], a[max]);
        if ( a[i] == missing )  missing--;
        else if ( a[i] < missing )  break (3 分);
    }
    return missing;
}
```

评测结果：答案正确（6 分）

序号	结果	得分
0	答案正确	3
1	答案正确	3

函数题总分：8 得分：8

6-1

Shortest Path [2]

Write a program to find the weighted shortest distances from any vertex to a given source vertex in a digraph. It is guaranteed that all the weights are positive.

Format of functions:

```
void ShortestDist( MGraph Graph, int dist[], Vertex S );
```

where `MGraph` is defined as the following:

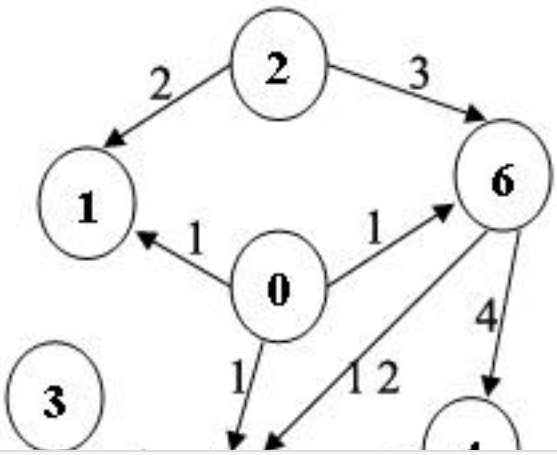
```
typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;
    int Ne;
    WeightType G[MaxVertexNum][MaxVertexNum];
};
typedef PtrToGNode MGraph;
```

The shortest distance from V to the source S is supposed to be stored in dist[V]. If V cannot be reached from S, store -1 instead.

Sample program of judge:

```
#include <stdio.h>
#include <stdlib.h>

typedef enum {false, true} bool;
#define INFINITY 1000000
#define MaxVertexNum 10  /* maximum number of vertices */
typedef int Vertex;      /* vertices are numbered from 0 to MaxVertexNum-1 */
typedef int WeightType;

typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;
    int Ne;
    WeightType G[MaxVertexNum][MaxVertexNum];
};
typedef PtrToGNode MGraph;

MGraph ReadG(); /* details omitted */

void ShortestDist( MGraph Graph, int dist[], Vertex S );

int main()
{
    int dist[MaxVertexNum];
    Vertex S, V;
    MGraph G = ReadG();

    scanf("%d", &S);
    ShortestDist( G, dist, S );

    for ( V=0; V<G->Nv; V++ )
        printf("%d ", dist[V]);

    return 0;
}

/* Your function will be put here */
```

Sample Input (for the graph shown in the figure):




```
7 9
0 1 1
0 5 1
0 6 1
5 3 1
2 1 2
2 6 3
6 4 4
4 5 5
6 5 12
2
```

Sample Output:

```
-1 2 0 13 7 12 3
```

代码

```
void ShortestDist( MGraph Graph, int dist[], Vertex S )
{
       int nv = Graph->Nv;
       int visited[nv];
       for(int i = 0; i < nv; i++)
       {
              dist[i] = INFINITY;
              visited[i] = 0;
       }
       dist[S] = 0;
       visited[S] = 1;
       for(int i = 0; i < nv-1; i++)
       {
              for(int j = 0; j < nv; j++)
                     if(!visited[j]&&dist[j] > dist[S] +
Graph->G[S][j])
                            dist[j] = dist[S] + Graph->G[S][j];
              int min = INFINITY, min_i = -1;
              for(int j = 0; j < nv; j++)
                     if(!visited[j]&&dist[j]<min)
                     {
```

```
                for(int j = 0; j < nv; j++)
                        if(!visited[j]&&dist[j]<min)
                        {
                                min = dist[j];
                                min_i = j;
                        }
                if(min_i == -1)      break;
                S = min_i;
                visited[S] = 1;
        }
        for(int i = 0; i < nv; i++)
                if(visited[i] == 0)
                        dist[i] = -1;
}
```

评测结果：答案正确（8 分）

测试点	结果	得分	耗时	内存
0	答案正确	4	3 ms	256 KB
1	答案正确	3	3 ms	384 KB
2	答案正确	1	3 ms	256 KB

```
a.c: In function ‘ReadG’:
a.c:55:2: warning: ignoring return value of ‘scanf’, declared
with attribute warn_unused_result [-Wunused-result]
  scanf("%d", &Nv);   /* 读入顶点个数 */
  ^~~~~~~~~~~~~~~~
a.c:58:2: warning: ignoring return value of ‘scanf’, declared
with attribute warn_unused_result [-Wunused-result]
  scanf("%d", &(Graph->Ne));   /* 读入边数 */
  ^~~~~~~~~~~~~~~~~~~~~~~~~
a.c:63:7: warning: ignoring return value of ‘scanf’, declared
with attribute warn_unused_result [-Wunused-result]
       scanf("%d %d %d", &E->V1, &E->V2, &E->Weight);
       ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
a.c: In function ‘main’:
a.c:80:5: warning: ignoring return value of ‘scanf’, declared
with attribute warn_unused_result [-Wunused-result]
     scanf("%d", &S);
     ^~~~~~~~~~~~~~~
```