1、G

2、B

3.C

4.B

5·D

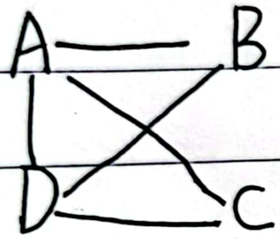
6.①:D ②B

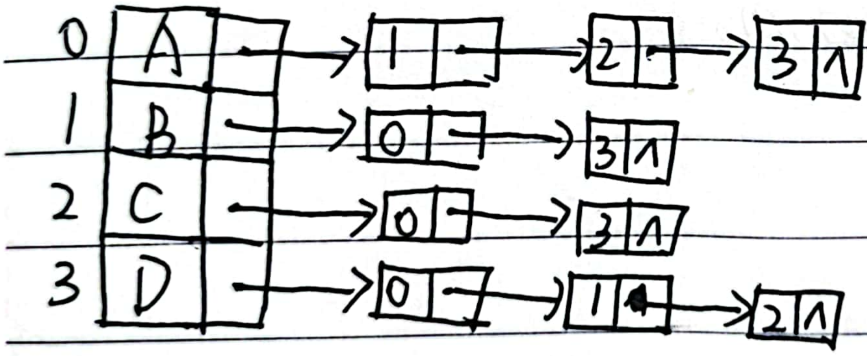
7.A

8.(1).D

(2).B

9.
(1).





(2). DFS: ABDC

BFS. ABCD

10. 18、

有两条：分别为

$A \xrightarrow{6} B \xrightarrow{1} E \xrightarrow{9} G \xrightarrow{2} I$ 和 $A \xrightarrow{6} B \xrightarrow{1} E \xrightarrow{7} H \xrightarrow{4} I$.

```
11. #include<stdio.h>
int main( )
{ int sig[7][7];
 for(int i=0 ; i<7;i++)
    for(int j=0;j<7;;++)
   { scanf("%d", &sig[i][j]);
     if(sig[i][j]==-1)
      sig[i][j]==∞;}
  int flag[7],dist[7];
 for(int i=0; i<7;i++)
     { flag[i]=1;
      dist[i]=sig[0][j];}
   flag[0]=0 ; int min=∞,m;
for(int k=0;k<6;k++)
 { min=∞;
   for(int i=0;i<7;i++)
    { if(dist[i]<min&& flag[i])
      { min=dist[i];
        m=i;}
   Flag[m]=0
  for(int i=0;i<7;i++)
   if(dist[m]+sig[m][i]<dist[i] && flag[i])
```

```
        dist[i]=dist[m]+sig[m][i];
                                  }
      return 0;                  }.
12. #include <stdio.h>
int main( )
{ int sig[5][5];
   for (int i=0;i<5;i++)
      for(int j=0;j<5;j++)
         { scanf("%d",&sig[i][j]);
           if(sig[i][j]==-1)
              sig[i][j]=∞;  }
   ~~For~~ int flag[5],dist[5],pre[5];
   for(int i=0;i<5;i++)
      {flag[i]=1;
        dist[5]=sig[0][i];}
      flag[0]=0
   for(int i=0;i<5;i++)
      if(~~sig~~ dist[i]<∞)
         pre[i]=0
   int min,m,sum=0;
 for (int k=0;k<4;k++)
   { min=∞;
```

```
for(int i=0;i<5;i++)
  if(dist[i]<min&&flag[i])
    {min=dist[i];
     m=i;}
  flag[m]=0;
  sum+=min;
for(int i=0;i<5;i++)
  if(sig[m][i]<dist[i]&&flag[i])
     {dist[i]=sig[m][i];
      pre[i]=m;}}
  return 0;};
```

(Prim算法).

```
#include<stdio.h>.
int main()
{ int sig[5][5];
   ........
  int vest[5];
 for(int i=0;i<5;i++)
   vest[i]=5; MST[4];
 Edge E[20]; int t=0;
for(int i=0;i<5;i++)
  for(int j=0;j<i;j++)
```

```
typedef struct
{ int from;
  int to;
  int weight;} Edge;
  if(sig[i][j]!=0 && sig[i][j]<∞)
        ~~E[t]~~+ {E[t].from=i;
                   E[t].to=j;
                   E[t].weight=sig[i][j]}
 qsort(E); //给E排序
 int count=0;
 int i=0,x,y;
while (count<4)
 {x=vest[E[i].from];
  y=vest[E[j].to];
  if(x!=y)
  {  MST[count++]=E[i]
    for(int j=0; j<5;j++)
       if (vest[j]==y)
            vest[j]=x;
   }
  i++;
  }
return 0;}
```

Kruskal 算法.