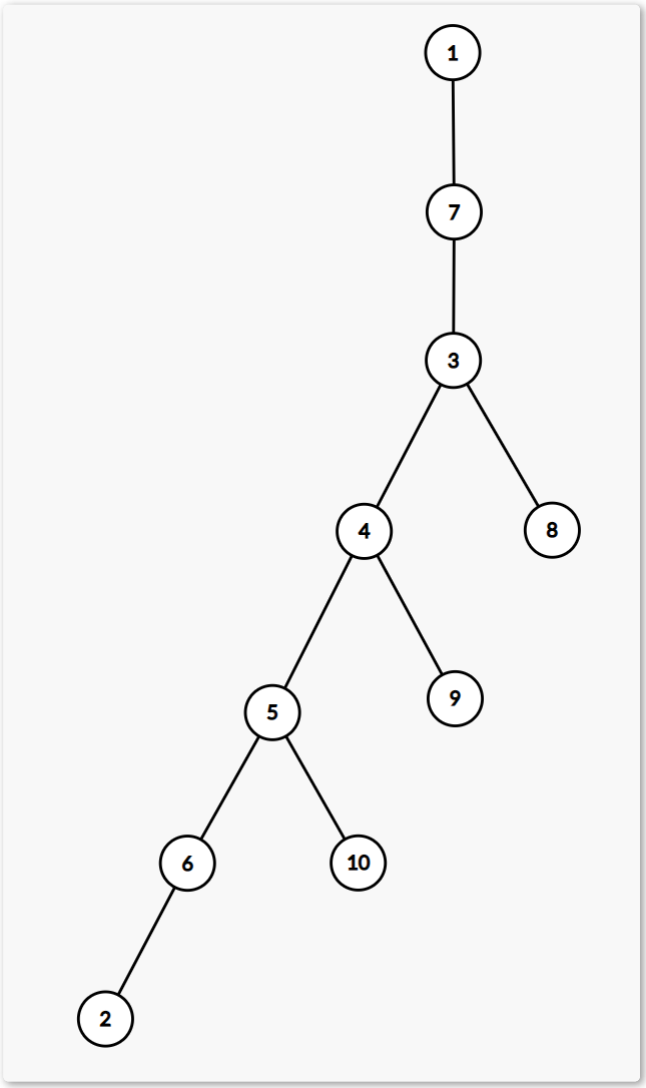
Homework9nd

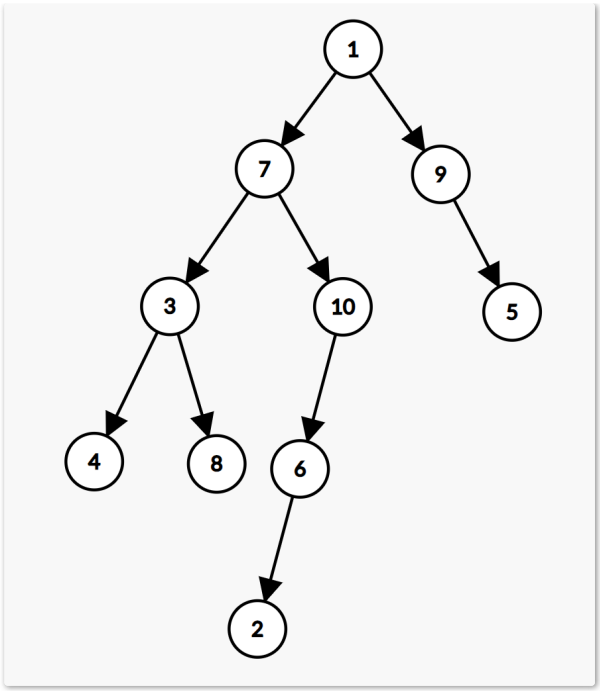
选择题

- 13：C
- 14：D

应用题

- 3：
  - 深度优先生成树
    - 

  - 广度优先生成树
    -



算法设计题

- 2:

```
#include<bits/stdc++.h>
using namespace std;
const int maxn=1e5+5;
int n,m,V;
struct Arc{
    int v;
    Arc* nxt;
    Arc():nxt(NULL){}
    Arc(int v):v(v),nxt(NULL){}
};
Arc *head[maxn],*tail[maxn],*curr[maxn];
void AddArc(int u,int v){
    Arc* tmp=new Arc(v);
    tail[u]->nxt=tmp;
    tail[u]=tmp;
    printf("tail %d =%d %p\n",u,v,tail[u]);
}
int stk[maxn],top;
bool vis[maxn];
int main(){
    cin>>n>>m>>V;
    int u,v;
    for(int i=1;i<=n;i++)head[i]=tail[i]=new Arc;
    for(int i=0;i<m;i++){
        cin>>u>>v;
```

```
            AddArc(u,v);
            AddArc(v,u);
        }
        vis[stk[top=1]=V]=1;
        curr[V]=head[V];
        while(top){
            int u=stk[top];
            if(curr[u]=curr[u]->nxt){
                int v=curr[u]->v;
                if(!vis[v]){
                    vis[stk[++top]=v]=1;
                    curr[v]=head[v];
                    printf("%d->%d\n",u,v);
                }
            }else top--;
        }return 0;
}
```

- **4:**

  ```
  #include<bits/stdc++.h>
  using namespace std;
  #define MaxVertexNum 30
  int vn,en;
  class MGraph{
      int n;
      char vexs[MaxVertexNum];
      bool edges[MaxVertexNum][MaxVertexNum];
  public:
      int getid(char v)const{
          for(int i=0;i<n;i++)if(vexs[i]==v)
              return i;
          exit(114514);
      }
      void Read(int vn,int en){
          n=vn;
          for(int i=0;i<vn;i++)cin>>vexs[i];
          char u,v;
          for(int i=0;i<en;i++){
              cin>>u>>v;
              edges[getid(u)][getid(v)]=edges[getid(v)]
  [getid(u)]=1;
          }
      }
      bool operator()(int u,int v)const{
  ```

```
        if(u<0||u>=n||v<0||v>=n)exit(114514);
        return edges[u][v];
    }
    char operator[](int id)const{
        return vexs[id];
    }
}G;
int stk[MaxVertexNum],top;
bool instk[MaxVertexNum];
void dfs(int u,int e){
    if(u==e){
        for(int i=1;i<=top;i++)printf("%c->",G[stk[i]]);
        printf("%c\n",G[e]);
        return;
    }
    instk[stk[++top]=u]=1;
    for(int v=0;v<vn;v++)if(!instk[v]&&G(u,v))
        dfs(v,e);
    top--;
    instk[u]=0;
}
int main(){
    cin>>vn>>en;
    G.Read(vn,en);
    for(int s=0;s<vn;s++)
        for(int t=s+1;t<vn;t++){
            printf("path from %c to %c:\n",G[s],G[t]);
            dfs(s,t);
        }
    return 0;
}
/*
5 5
a b c d e
a  b
b c
b d
c e
d e
*/
```