Seminarul 13 - Programare Dinamica

miercuri, 25 mai 2022 15:58

Subsir crescator maximal

scmax

Input: v[]

Output: l max a. i. E i1<i2<i3<..<il , a[i1]<a[i2]<…<a[il]

dp[i]= cel mai lung subsir crescator care se termina exact la poz i

Cod:

```
    functie(v){
        dp[1|x in [0..a.size()];
        ans=1;
        for(i=1;i<a.size();i++)
            for(j=0;j<i;j++){
                dp[i]=max(dp[i], dp[j]+1);
            }
            ans=max(ans, dp[i]);
        }
    return ans;
    }
```

Bancnote:

Input: S, B={b1, b2, b3, …, bk}

dp[i]=nr minim de bancnote necesar pentru a plati suma i

```
dp[i]=min(   | 1+dp[i-b1]
             | .
             | .
             | .
             | 1+dp[i-bk])
```

Cod:

```
    SMB(s, b){
        dp[]=infinit;
        dp[0]=0;
        for(i=1;i<=s;i++){
            mini=inf;
            for(j=0;j<b.size();j++){
                if(dp[i-b[j]]+1<mini)
                    mini=dp[i-b[j]]+1;
            }
            dp[i]=mini;
        }
        return dp[s];
    }
```

Problema mai speciala; suma maxima in arbore

dp0[nod]=suma max la subarbore, fara 'nod'
dp1[nod]=-/-, cu 'nod'

```
struct Node{
   int val, id;
   vector<Nod.c> children
}

dp0=[]; dp1=[];
DFS(root){
   if(root=NULL) return;
   dp0[root->id]=0; dp1[root->id]=root->val;
   for(child in root->children){
      DFS(child);
      dp1[root->id]+=dp0[child->id];
      dp0[root->id]+=max(dp0[child->id],dp1[child->id]);
   }
}
main(){
   DFS(root);
   print(max(dp0[root->id],dp1[root->id]);
}
```