34. Binary Search tree

#include<stdio.h>

#include<limits.h>

int min(int a,int b)

{

return(a<b)?a:b;

}

int sum(int freq[],int i,int j)

{

int s=0;

for(int k=i;k<=j;k++)

{

s+=freq[k];

}

return s;

}

int optimalBSTCost(int keys[],int freq[],int n)

{

int dp[n+1][n+1];

for(int i=0;i<n;i++)

{

dp[i][i]=freq[i];

}

for(int length=2;length<=n;length++)

{

for(int i=0;i<=n-length+1;i++)

{

int j=i+length-1;

dp[i][j]=INT\_MAX;

for(int k=i;k<=j;k++)

{

int cost=((k>i)?dp[i][k-1]:0)+((k<j)?dp[k+1][j]:0)+sum(freq,i,j);

dp[i][j]=min(dp[i][j],cost);

}

}

}

return dp[0][n-1];

}

int main()

{

int n;

int keys[n];

int freq[n];

printf("Enter the number of keys: ");

scanf("%d",&n);

printf("Enter the keys:\n");

for(int i=0;i<n;i++)

{

scanf("%d",&keys[i]);

}

printf("Enter the frequencies:\n");

for(int i=0;i<n;i++)

{

scanf("%d",&freq[i]);

}

int cost=optimalBSTCost(keys,freq,n);

printf("Minimum cost of Optimal Binary Search Tree: %d\n", cost);

return 0;

}

