### 木棍dfs剪枝

#include <cstdio>  
#include <algorithm>  
#include <cstring>  
using namespace std;  
const int maxn = 105;  
int sticks[maxn];  
int vis[maxn]; /\*记录每根木棍的访问\*/  
int n; /\*多组输入\*/  
int max\_len; /\*假设最长木棍长度\*/  
int max\_cnt; /\*最大木棍数量\*/  
  
/\*  
 \* 正在拼接第stick根原始木棍(已经拼好了stick-1根)  
 \* 第stick根木棍的当前长度为now\_len  
 \* 拼接到第stick根木棍中的上一根小木棍为last  
 \*/  
bool DFS(int stick, int now\_len, int last) {  
 /\*所有原始木棍都已拼好，搜索成功\*/  
 if (stick > max\_cnt)  
 return true;  
 /\*第stick根木棍已经拼好，去拼下一根\*/  
 if (now\_len == max\_len)  
 return DFS(stick+1, 0, 1);  
  
 /\*记录尝试向当前原始木棍拼接的最近的失败的木棍长度\*/  
 int record = 0;  
 for (int i = last; i <= n; ++i)  
 if (!vis[i] && now\_len + sticks[i] <= max\_len && record != sticks[i]) {  
 vis[i] = 1;  
 if (DFS(stick, now\_len + sticks[i], i + 1))  
 return true;  
 record = sticks[i];  
 vis[i] = 0; /\*还原现场\*/  
 /\*贪心，再用1根木棍恰好拼完当前原始木棍必然比再用若干根木棍拼完更好\*/  
 if (now\_len == 0 || now\_len + sticks[i] == max\_len)  
 return false;  
 }  
 /\*所有分支均尝试过，搜索失败\*/  
 return false;  
}  
  
int main()  
{  
 while (scanf("%d", &n) && n) {  
 int sum = 0;  
 int val = 0;  
 for (int i = 1; i <= n; ++i) {  
 scanf("%d", &sticks[i]);  
 sum += sticks[i];  
 val = max(sticks[i], val);  
 }  
 sort(sticks+1, sticks+1+n);  
 reverse(sticks+1, sticks+1+n);  
 for (int max\_len = val; max\_len <= sum; ++max\_len) {  
 if (sum % max\_len)  
 continue;  
 max\_cnt = sum / max\_len;  
 memset(vis, 0, sizeof(vis));  
 if (DFS(1, 0, 1))  
 break;  
 }  
 printf("%d\n", max\_len);  
 }  
 return 0;  
}