Hello World ! ---vijos

DP 背包 搭建双塔

题意：N块水晶的高度Hi，判断能否用这些水晶搭建成一座有同样的高度的双塔，如果能，则输出所能搭建的双塔的最大高度，否则输出“Impossible”。

思路：判定性问题

f[j][k]表第一座塔高为j，第二座塔高为k是否可能

f[j][k] = f[j-a[i]][k] || f[j][k-a[i]]

讨论里有人用二分过，应该是二分+背包判定吧

要加sort和min(sum,m)优化，不然会超时

线性DP 滑雪

因为初始化WA了一次，要初始化为1，而不是无穷小

思路：按高度h排序一遍，记录每个(x,y)在排序后的位置，然后按高度序列扫一遍，扫到(x,y)时，

能从(x-1,y),(x+1,y),(x,y-1),(x,y+1)状态转移，根据坐标找到相应的位置即可

区间DP输出解 好题

题意：给定一个正整数序列a(1)，a(2)，...，a(n),(1<=n<=20).不改变序列中元素在序列中的位置，把它们相加，并用括号记每次加法所得和，称为中间和.

现在要添上n-1对括号，加法运算依括号顺序进行，得到n-1个中间和，求出使中间和之和最小的添括号方法。

思路：裸区间DP，主要是输出解...搞着搞着就出来了=\_= 其实可以一遍AC的，数据没有special judge

遇到2 2 2这种数据要输出((2+2)+2)而不能是(2+(2+2))

输出解：output(i,j) {

k = find\_correct\_mid(i, j);

output(1, k), output(k+1, j);

}

中青局 : 哈夫曼编码 huffman 实现 好题

题意：一串字符串，用0/1编码，使得编码最短且具有唯一性，输出编码长度

思路：huffman tree

大牛们的实现好酷炫，学到了~

摘：fjxmlhx大牛说得好：

只要编码满足任意两个i j i的编码不是j的编码的前缀就行，于是考虑构造一棵trie满足节点是01，且每个叶子都对应了一个字母，且根到叶子的距离和\*对应单词权最小..

接下来考虑每一棵可行的树，实际上等于N棵只有一个叶子节点且树权为单词权的树，按照下面规则合并得到，规则是：

1：由给定的n个权值构造n棵树只有一个叶子结点的二叉树，得到一个二叉树的合F； 2：再F中选取二棵二叉树作为左，右子树构造一棵新的二叉树，这棵树的根结点的权值为左右子树根结点权值之和。 3：在集合F中删除作为左右子树的二棵二叉树，并将新建的二叉树加入到集合F中。 4：重复2 ,3两步，当F中只剩下一棵二叉树时，结束

可以看出，如果能找到使得最后代价最小的合并顺序，那么就他就一定是答案，可以发现最小的合并顺序求法就是合并果子，结束。

\*\*\* 一棵trie满足节点是01，且每个叶子都对应了一个字母，且根到叶子的距离和\*对应单词权最小\*\*\*

int solve()

{

memset(cnt, 0, sizeof(cnt));

for(int i = 0; i < len; i++) {

str[i] = str[i] == '\_' ? 26 : str[i] - 'A';

cnt[str[i]]++;

}

while(!Q.empty()) Q.pop();

for(int i = 0; i < 27; i++) if(cnt[i]) Q.push(cnt[i]);

if(Q.size() == 1) { return Q.top() ; }

int ans = 0;

while(Q.size() > 1) {

int x = Q.top(); Q.pop(); int y = Q.top(); Q.pop();

ans += x + y;

Q.push(x+y);

}

return ans;

}

int main()

{

while(scanf("%s", str), strcmp(str, "END\0")) {

len = strlen(str);

int ans = solve();

printf("%d %d %.1f\n", len\*8, ans, len\*8.0/ans);;

}

return 0;

}