

Ajude o Autor do Problema

Contest Local, Universidade de Ulm  Alemanha**Timelimit: 1**

Preparar um problema para uma competição de programação leva muito tempo. Você não somente tem que escrever a descrição e a solução de um problema, mas você também tem que criar arquivos de entrada difíceis. Nesse problema, você terá a chance de ajudar o autor de problemas a criar algumas entradas para um determinado problema.

Para isso vamos selecionar o problema que não foi resolvido durante uma competição local no ano passado. O problema era encontrar a árvore binária de busca ótima, dado a probabilidade de que certos nós são acessados. Seu trabalho será: dado a árvore binária de busca ótima desejada, descubra algumas probabilidades de acesso para que esta seja a única e melhor árvore. Não se preocupe se você não leu o problema no ano passado, pois todas as definições exigidas foram colocadas abaixo.

Vamos definir uma **árvore binária de busca** indutivamente da seguinte forma:

- Uma árvore vazia que não possui nenhum nó é uma árvore binária de busca;
- Cada árvore binária de busca não vazia possui uma raiz, que é um nó identificado por um inteiro e duas sub-árvores uma a direita e outra a esquerda da raiz;
- A sub-árvore da esquerda não contém nenhum nó com rótulo \geq que o rótulo da raiz;
- A sub-árvore da direita não contém nenhum nó com rótulo \leq que o rótulo da raiz.

Dado tal árvore de busca binária, o seguinte **procedimento de busca** pode ser usado para localizar um nó na árvore:

Comece com a raiz. Compare o rótulo do nó atual com o rótulo do nó desejado. Se forem iguais, você encontrou o nó correto. Caso contrário, se o rótulo desejado for menor busque na sub-árvore da esquerda, senão, busca na sub-árvore da direita.

O **custo de acesso** para localizar um nó é o número de nós que você visitou até encontrar o nó certo. Uma **árvore binária de busca ótima** é uma árvore com o mínimo custo esperado.

Entrada

O entrada contém vários casos de teste. Cada caso inicia com um inteiro n ($1 \leq n \leq 50$), que é o número de nós de uma árvore binária de busca ótima. Para simplificar, os valores dos nós serão inteiros de 1 a n . A seguir temos n linhas que descrevem a estrutura da árvore. A i -ésima linha contém os valores dos nós da sub-árvore esquerda e direita com valor i (ou -1 para uma árvore vazia). Você pode assumir que a entrada sempre conterá uma árvore binária de busca válida.

O último caso é seguido por um zero.

Note que o primeiro caso de teste no exemplo de entrada descreve uma árvore parecida com:

```

  2
 / \
1   3
```

Saída

Para cada caso de teste, escreva uma linha contendo a frequência de acesso para cada nó em ordem crescente de valores dos nós. Para evitar problemas de precisão, as frequências devem ser escritas como números inteiros, significando que a probabilidade de acesso de um nó será a frequência dividida pela soma de todas as

frequências. Certifique-se que você não escreverá qualquer inteiro maior que $2^{63} - 1$ (o valor máximo que cabe no tipo long long para C/C++ ou long para Java). Caso contrário, você poderá produzir uma solução qualquer sem a garantia de que é exatamente uma árvore binária de busca ótima: a árvore binária de busca dada na entrada.

Exemplo de Entrada	Exemplo de Saída
3 -1 -1 1 3 -1 -1 10 -1 2 -1 3 -1 4 -1 5 -1 6 -1 7 -1 8 -1 9 -1 10 -1 -1 0	1 1 1 512 256 128 64 32 16 8 4 2 1