

# Construção de Procura Binária de Heap

Contest Local, Universidade de Ulm  Alemanha**Timelimit: 2**

Leia o enunciado do problema G para as definições sobre heaps. A seguir nós definimos a terminologia básica de heaps. Uma heap é uma árvore cujos nós internos tem, cada um, uma prioridade (definida por um número) sendo que a prioridade de cada nó interno é menor que a prioridade de seu nó-pai. Como consequência, a raiz será o nó de maior prioridade da árvore. Isso é uma das razões pelas quais heaps podem comumente ser usadas para a implementação de filas de prioridade e para ordenações.

Uma árvore binária na qual cada nó interno tem ambos um rótulo e uma prioridade, e é tanto uma árvore binária de busca com atenção para rótulos; quanto uma fila com atenção para prioridades, é chamada de treap (árvore-heap). A sua tarefa é: Dado um conjunto de pares de rótulos e prioridades, com rótulos únicos e prioridades únicas, construir uma treap com essas informações.

## Entrada

A entrada contém vários casos de teste. Cada caso de teste começa com um inteiro  $n$ . Você pode assumir que  $1 \leq n \leq 50000$ . Então segue  $n$  pares de strings e números rótulo<sub>1</sub>/prioridade<sub>1</sub>, ..., rótulo<sub>n</sub>/prioridade<sub>n</sub>. As strings são não-nulas e em caixa-baixa, e os números são inteiros não-negativos. O último caso de teste é seguido por um zero.

## Saída

Cada linha de cada caso de teste deve conter uma treap com os nós especificados. Uma treap é impressa como (<Sub-treap da Esquerda><Rótulo>/<Prioridade><Sub-treap da Direita>) As sub-treaps são impressas recursivamente e omitidas se forem folhas.

Exemplo de Entrada	Exemplo de Saída
7 a/7 b/6 c/5 d/4 e/3 f/2 g/1 7 a/1 b/2 c/3 d/4 e/5 f/6 g/7 7 a/3 b/6 c/4 d/7 e/2 f/5 g/1 0	(a/7(b/6(c/5(d/4(e/3(f/2(g/1))))))) ((((((a/1)b/2)c/3)d/4)e/5)f/6)g/7) ((a/3)b/6(c/4))d/7((e/2)f/5(g/1))