

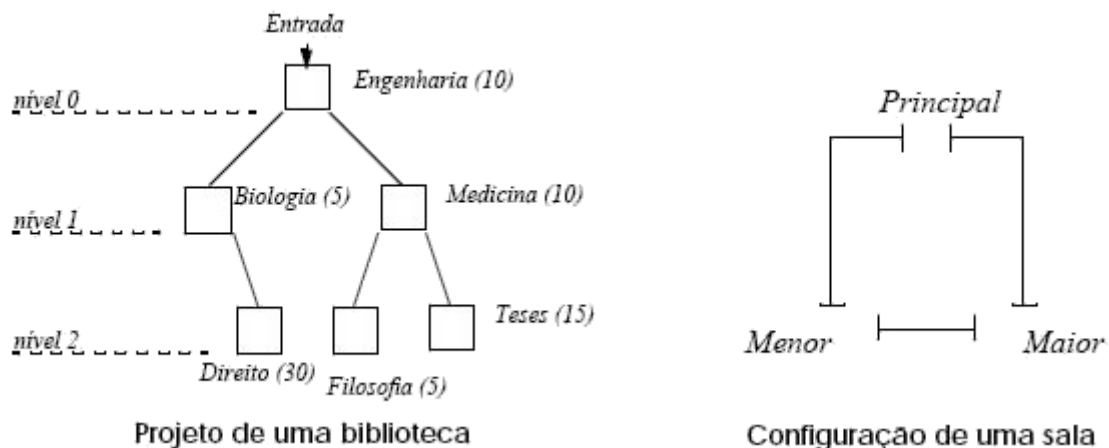
D. Biblioteca Ótima

Time limit: 0.181s

Memory limit: 1536 MB

O arquiteto Otávio B. Ignácio, da firma OBI Arquitetura Ltda, está criando um novo modelo para o projeto arquitetural de bibliotecas. Ele constatou que um visitante poderia ter que andar por toda a biblioteca procurando pela seção à qual pertence o livro desejado caso não houvesse alguma ordem na localização das seções (encontrar o livro dentro da seção é mais fácil porque eles normalmente estão ordenados). O novo modelo procura resolver o problema, de forma que um visitante possa ir mais diretamente para a seção desejada, sem ter que passar por toda a biblioteca.

Pela sua idéia, cada seção está associada a uma única sala e vice-versa. As salas possuem até três portas com os nomes de *Principal*, *Menor* e *Maior*, conforme a figura abaixo, e as portas *Menor* e *Maior* de uma sala estão sempre associadas à porta *Principal* de uma outra sala. Além disso, seu modelo segue a seguinte propriedade: toda seção que se pode chegar através da porta *Menor* de uma determinada sala possui nome necessariamente menor do que o nome da seção atual; da mesma forma, toda seção que se pode chegar através da porta *Maior* possui nome necessariamente maior. A ordem utilizada para comparar os nomes das seções pode ser a mesma ordem utilizada em dicionários (ordem lexicográfica). Note que toda sala tem uma porta *Principal*, mas a existência das portas *Menor* e *Maior* depende da existência de seções adjacentes.



Desta forma, quando um visitante entra em uma sala por sua porta *Principal*, ele compara o nome da seção desejada com o nome da seção correspondente à sala. Se o nome da seção desejada for menor, o visitante segue seu caminho pela porta de nome *Menor*; se for maior, segue pela porta de nome *Maior*. Obviamente, se os dois nomes foram iguais, significa que ele encontrou a seção desejada.

Um amigo bibliotecário sugeriu que as seções mais procuradas deveriam ficar mais próximas da entrada principal, de forma a diminuir a distância média necessária para uma pessoa encontrar a seção desejada. No entanto, Otávio sabia que seu modelo de busca restringia a topologia não podendo-se simplesmente colocar as seções mais visitadas próximas à entrada sem considerar sua ordem relativa. Utilizando-se do número de visitas de cada seção em um ano como custo de um nó, ele definiu que o custo total de uma topologia seria dado pela soma total do custo de cada nó multiplicado pelo número de seções intermediárias no caminho desde a entrada da biblioteca até a sala desejada. Este último valor é equivalente ao nível de uma sala, conforme mostrado na figura. Sendo assim o custo da topologia adotada na figura é 115. Substituindo a seção de Biologia pela seção de Direito e fazendo a primeira acessível pela porta *Menor* da segunda geraria uma outra topologia de custo 90, para o mesmo exemplo de biblioteca.

Otávio concluiu que a topologia de custo total mínimo representa a melhor distribuição de seções em uma biblioteca do seu modelo. No entanto, Otávio calculou manualmente este valor para projetos pequenos e não sabe como resolver o problema em projetos maiores.

Tarefa

Você foi contratado para desenvolver um programa que calcule o custo total mínimo de uma biblioteca dentre todas as topologias possíveis, dadas as frequências de acesso às suas seções.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro N que indica o número de seções da biblioteca. As seções são identificadas por inteiros de 1 a N . A segunda linha do conjunto de teste contém N inteiros entre 0 e 100 , representando as frequências de acesso das seções $1, 2, 3, \dots$ e N , respectivamente. O final da entrada é indicado por $N = 0$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n ", onde n é numerado a partir de 1 . A segunda linha deve conter o custo total mínimo para a topologia indicada, calculado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
1
5
3
10 10 10
3
5 10 20
0
```

Saída:

```
Teste 1
0

Teste 2
20

Teste 3
20
```

Restrições

$0 \leq N \leq 60$ ($N = 0$ apenas para indicar o fim da entrada)

$0 \leq \text{Frequências} \leq 100$

Olimpiada Brasileira de Informatica 2002 - Seletiva