SCC603 – Algoritmos e Estruturas de Dados II

Prof.ª Maria Cristina Ferreira de Oliveira

1º sem. 2014

Trabalho 1 – Tesouro

Data de Entrega: 24/04/2014

Introdução

Um famoso explorador descobriu uma região repleta de labirintos, dentro dos quais sabe-se que há um valioso tesouro. Estes labirintos, porém, têm algumas peculiaridades.

Cada labirinto é composto por salas, tendo cada sala uma parte (não necessariamente igual) do tesouro, a qual pode ser totalmente coletada ao se visitar a sala.

As salas são ligadas entre si através de corredores. Cada corredor liga uma sala origem a uma sala destino e possui duas portas, uma em cada extremidade. Inicialmente, a porta da sala de origem está aberta, enquanto a porta da sala de destino está fechada. Ao passar pela porta da sala de origem, esta porta é fechada, e a porta da sala de destino é aberta. Após percorrer o corredor e passar pela porta da sala de destino, esta porta é fechada. Uma vez que isso ocorreu, as portas permanecem nessa posição para sempre, de modo que o corredor só pode ser utilizado uma vez, e apenas da sala de origem para a sala de destino.

Além disso, há uma sala especial (sem tesouro) que serve de entrada para o labirinto, da qual partem corredores (como os descritos anteriormente) para todas as outras salas. Da mesma forma, há uma sala especial que serve de saída para o labirinto, para a qual convergem corredores vindos de todas as outras salas. As portas de entrada para a sala de entrada e de saída da sala de saída são como as portas dos corredores, fechando-se ao serem ultrapassadas. Desse modo, só se pode entrar e sair de um labirinto uma única vez.

Mais ainda, o labirinto é construído de tal forma que, uma vez tendo saído de uma sala, não há corredores que levem de volta a esta mesma sala, não sendo possível, então, visitá-la mais de uma vez. Ou seja, não há nenhuma sequência de corredores que podemos percorrer tal que uma sala é visitada mais de uma vez.

Levando-se em conta, então, as características desses labirintos, é pedido que você faça um programa que, dados a quantidade de tesouro encontrada em cada sala e os corredores que as ligam, calcule o máximo de tesouro que pode ser coletado de um determinado labirinto.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste corresponde a um labirinto, descrevendo suas salas e seus corredores.

A primeira linha de um caso de teste contém um único número inteiro, n (1 <= n <= 1000), correspondendo ao número de salas do labirinto, numeradas de 1 a n. A seguir, há n linhas, tendo a i-ésima destas linhas informações sobre a sala de número i.

As informações sobre uma sala estão separadas por espaços e na seguinte ordem: um número inteiro q(0 <= q <= 10000), a quantidade de tesouro encontrada na sala; um número inteiro p (0 <= p < n), a quantidade de corredores que têm origem na sala i; e p números inteiros, cada um entre 1 e n, com os números das salas de destino dos corredores que têm origem na sala sendo descrita.

Não são descritas as salas de entrada e de saída do labirinto e os corredores que as ligam às outras salas, uma vez que são estruturas especiais do labirinto que dispensam descrição.

A entrada termina quando n = 0.

Saída

Para cada caso de teste, seu programa deve imprimir, em uma única linha, o máximo de tesouro que pode ser coletado das salas do labirinto.

Exemplo

Entrada:

```
3
7 1 3
10 0
8 0
3
7 1 3
20 0
8 0
6
100 0
300 0
100 1 1
0 1 2
100 1 3
1 1 4
0
```

Saída:

```
15
20
301
```

Outras Informações Importantes

- O trabalho deve ser feito em duplas, sendo que ambos os membros da dupla devem submetê-lo através do Sistema de Submissão de Programas (SSP) na disciplina **SCC603 – Algoritmos e Estruturas de Dados 2 (2014/1) Turma A.** .
- Todas as submissões são checadas para evitar cópia/plágio/etc. Portanto, evite problemas e implemente o seu próprio código.
- Comente o seu código com uma explicação rápida do que cada função, método ou trecho importante de código faz (ou deveria fazer). Os comentários serão checados e valem nota.
- Mantenha a modularização entre código e dados, ou seja, divida o código explicitamente entre estruturas de dados e algoritmos de manipulação. A modularização será checada e vale nota.
- Entradas/saídas devem ser lidas/escritas a partir dos dispositivos de entrada e saída padrões, logo são suficientes as funções **printf()** e **scanf()**. Para testar o programa fora do SSP, pode-se usar redirecionamento de arquivos. Para isso utilize os operadores **<** e **>**, como no seguinte exemplo:

```
# ./trab1 < entrada.txt > saida.txt
```