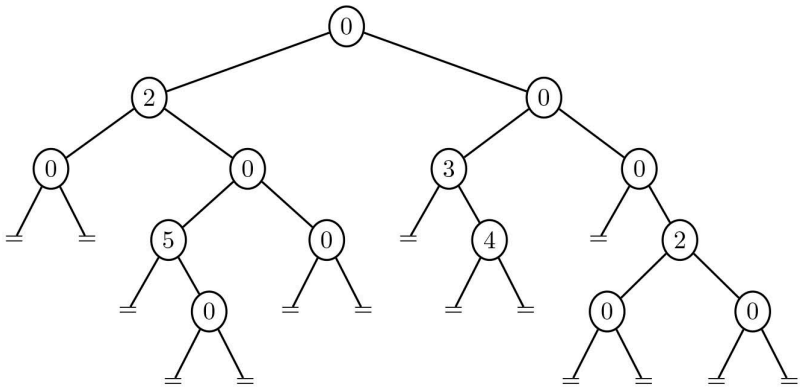
Excursionistas atrapados en una montaña

Durante el fin de semana varios grupos de excursionistas han intentado subir a una montaña. Debido a las condiciones meteorológicas han tenido que desistir en su intento y se encuentran atrapados en diversos puntos de la falda de la montaña. Las rutas que suben a la montaña se estructuran en forma de árbol binario. De la base de la montaña parten muchas rutas que se van juntando (una o dos) en diversas intersecciones hasta llegar solo uno o dos caminos a la cima. Se tiene la localización de cada grupo (se encuentran todos ellos en intersecciones de caminos) y su número de componentes.

Se están organizando equipos de rescate para ir a buscarlos. Para facilitar el rescate partirá un equipo del punto de la base de la montaña más cercano a cada grupo atrapado en la parte baja. Los equipos luego irán subiendo por los caminos de la montaña rescatando a los grupos que se encuentran más cerca de la cima. Los equipos nunca bajan para buscar a grupos de excursionistas. Por lo tanto se necesitan tantos equipos como grupos hay que no tengan otro grupo en ninguna de las rutas que suben hasta ese punto.



En el ejemplo, tenemos 5 grupos de excusionistas perdidos en la montaña. Se necesita un equipo de rescate para el grupo de 5, este equipo rescatará también al grupo de 2 que se encuentra en la ruta entre el 5 y la cima. Otro equipo rescatará al grupo de 4 y después al grupo de 3 que se encuentra encima. Un último equipo rescatará al grupo de 2 que se encuentra a la derecha. Si a un grupo lo pudieran rescatar dos equipos, uno por cada camino, es indiferente cual de ellos lo rescata.

Requisitos de implementación.

La función que calcula el número de equipos de rescate y el número de excursionistas atrapados en la ruta con mayor número de excursionistas debe ser externa a la clase *Arbin*. La función tendrá un parámetro de entrada correspondiente al árbol binario y devolverá una estructura con dos campos que representan el número de equipos y el número de excursionistas de la ruta mas cargada.

struct solucion {int numEquipos; int numExRuta;};

La solución debe tener un coste lineal respecto al número de nodos del árbol.

Entrada

La entrada comienza con el número de casos de prueba. Para cada caso se muestra el recorrido en preorden del árbol incluyendo los punteros a nulo en una línea.

Los valores de los nodos serán números enteros. Los valores a cero indican que no hay grupo de excursionistas en esa intersección, los valores distintos de cero indican el número de excursionistas de esa intersección. Los punteros a nulo de las hojas se representan con el valor -1.

Salida

Para cada caso de prueba se escribirá el número de equipos de rescate necesarios para socorrer a todos los excursionistas seguido del número de excursionistas que se encuentran atrapados en la ruta que tiene un mayor número de excursionistas. Si el árbol es vacío tendremos cero equipos y cero excursionistas.

Entrada de ejemplo

```
2
0 2 0 -1 -1 0 5 -1 0 -1 -1 0 -1 -1 0 3 -1 4 -1 -1 0 -1 2 0 -1 -1 0 -1 -1
1 0 3 -1 -1 -1 0 4 -1 0 -1 -1 0 -1 -1
```

Salida de ejemplo

```
3 7
2 5
```

Autor: Isabel Pita.