
Nombre de nodes del camí descendent més llarg amb el mateix valor X57266_ca

Implementeu una funció **RECURSIVA** que, donat un arbre binari de naturals positius, retorna el nombre de nodes del camí descendent més llarg a on tots els valors dels nodes son idèntics. En altres paraules, la funció retorna el natural N més gran tal que existeix una parella de nodes (u, v) de l'arbre, on u és antecessor de v , tots els nodes del camí de u a v contenen el mateix valor, i el nombre de nodes d'aquest camí és N . Aquesta és la capcelera:

```
// Pre: Tots els valors de t son naturals positius.
// Post: Retorna el nombre de nodes del camí descendent més llarg i amb el mate
int maxLengthConstantPath(BinTree<int> t);
```

Aquí tenim un exemple de paràmetre d'entrada de la funció i la corresponent sortida:

```
maxLengthConstantPath(      1      ) = 2
                        |
                    -----
                   |         |
                   3         2
                           |
                           -----
                          |
                          3
                          |
                          -----
                         |         |
                         1         3
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc`, `BinTree.hh`, `maxLengthConstantPath.hh`. Us falta crear el fitxer `maxLengthConstantPath.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Només cal que pugeu `maxLengthConstantPath.cc` al jutge.

Entrada

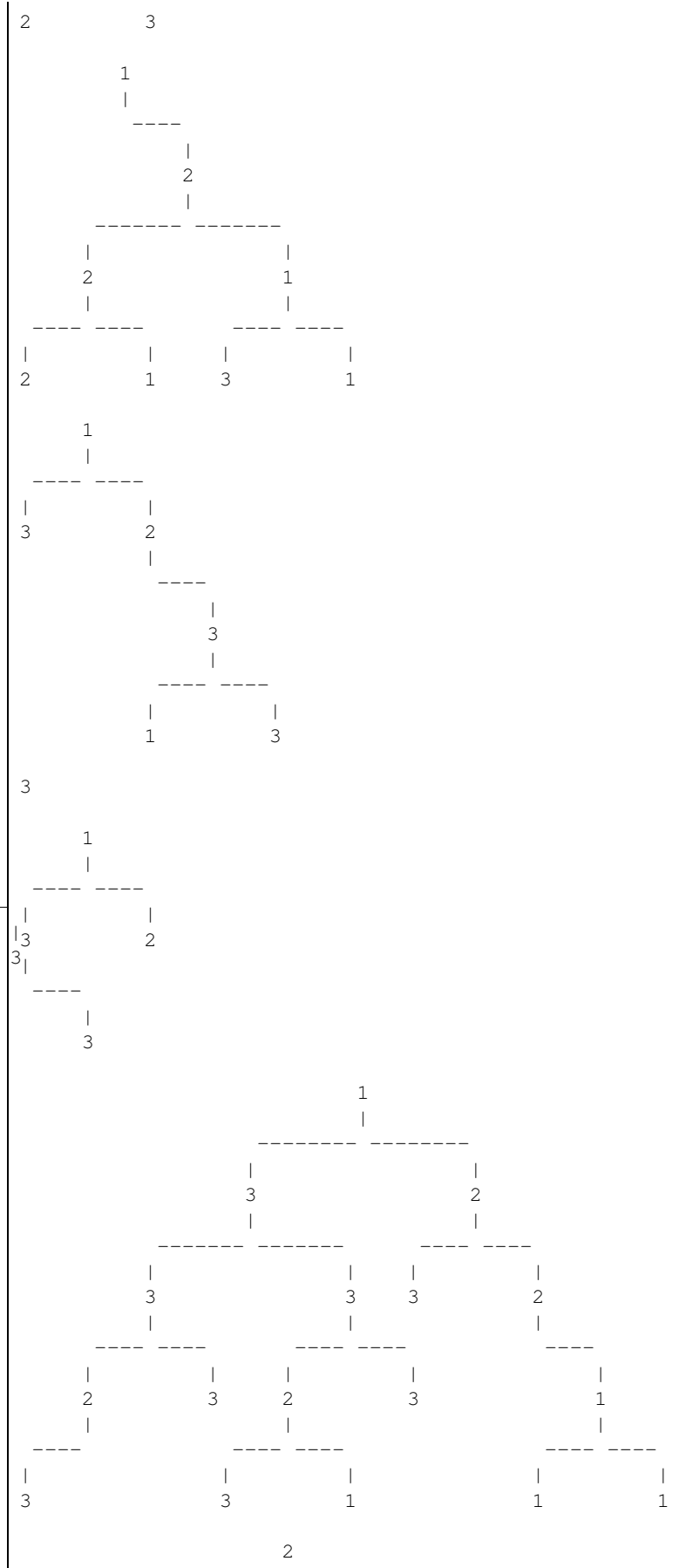
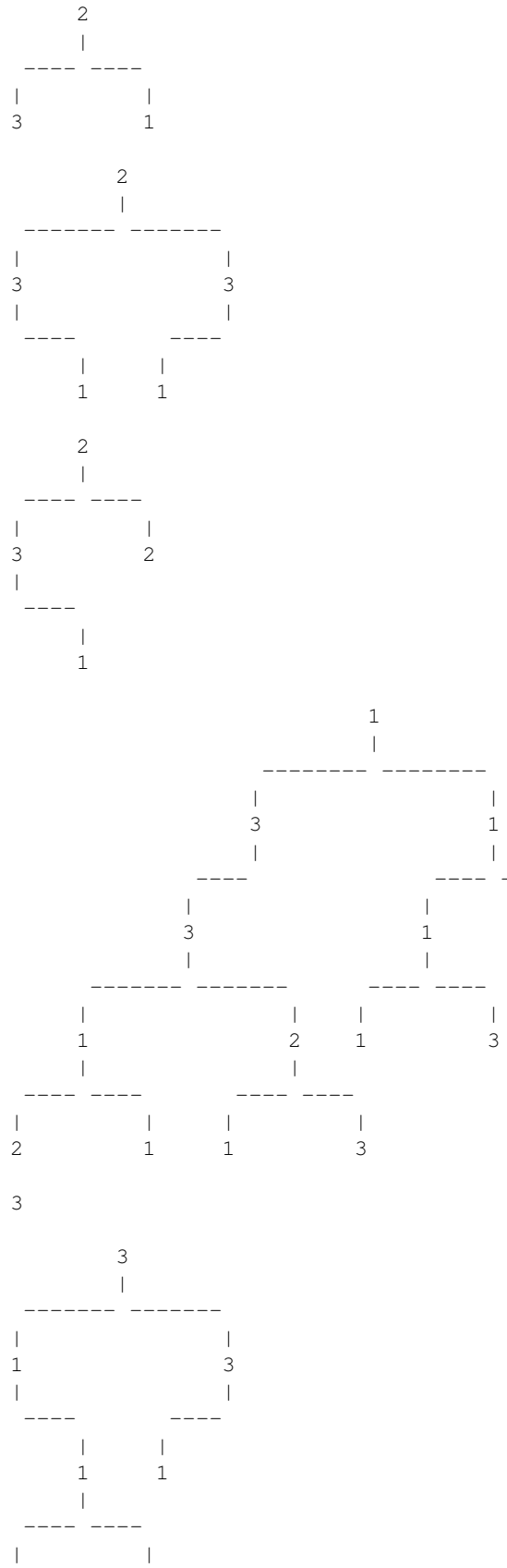
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `IN-LINEFORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

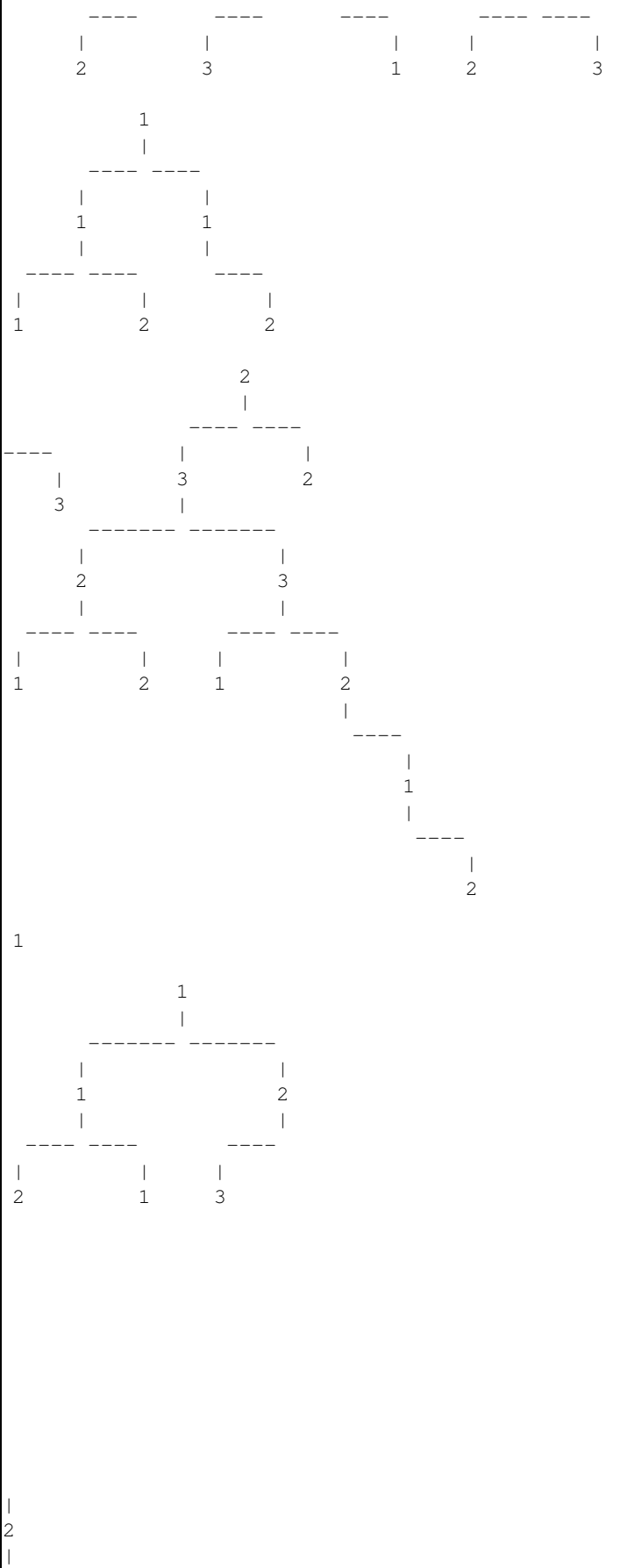
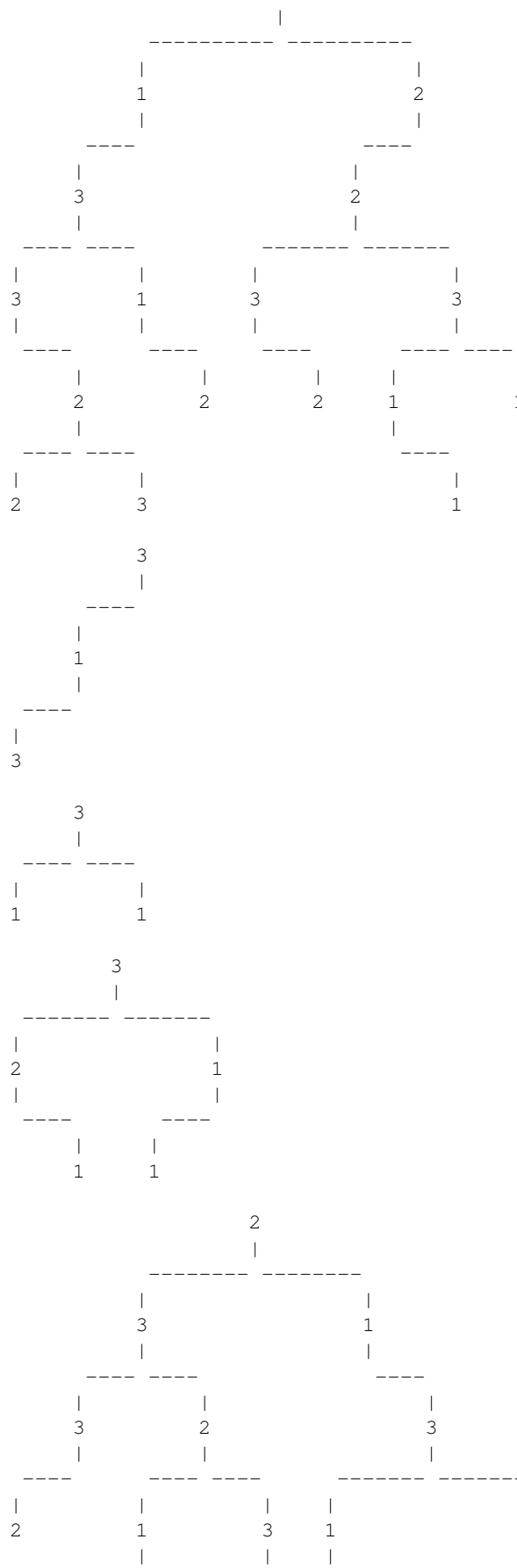
Sortida

Per a cada cas, la sortida conté el corresponent màxim nombre de nodes. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquest valor. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

VISUALFORMAT





Exemple de sortida 1

1
1
2
4
1
2
3
2
1

2
3
3
1
1
2
2
3
2
1
3

Exemple d'entrada 2

```
INLINEFORMAT
2(3,1)
2(3(,1),3(1,))
2(3(,1),2)
1(3(3(1(2,1),2(1,3)),),1(1(1,3),3))
3
3(1(,1(2,3)),3(1,))
1(,2(2(2,1),1(3,1)))
1(3,2(,3(1,3)))
3
1(3(,3),2)
1(3(3(2(3,),3),3(2(3,1),3)),2(3,2(,1(1,1)))3)
2(1(3(3(,2(2,3)),1(,2)),),2(2(3(,2),3(1(,1),1(,3))),))
3(1(3,))
3(1,1)
3(2(,1),1(1,))
2(3(3(2,),2(1(2,),3(3,))),1(,3(1(,1),2(2,3)))
1(1(1,2),1(,2))
2(3(2(1,2),3(1,2(,1(,2))))),2)
1
1(1(2,1),2(3,))
```

Exemple de sortida 2

1
1
2
4
1
2
3
2
3
3
2
3
1
2
2
2
2
1
2
2
2
1
3

Observació

Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autor : PRO2

Generació : 2024-02-14 00:18:23

© Jutge.org, 2006–2024.

<https://jutge.org>