

Aquest exercici explora la generació exhaustiva de multiconjunts.

1. Feu una funció `multisets1 :: Int → Int → [[Int]]` que, donats dos naturals n i x , retorni tots els multiconjunts que es poden formar amb $[1..n]$, de manera que cada nombre aparegui com a molt x vegades.
2. Feu una funció `multisets2 :: Int → Int → Int → [[Int]]` que, donats tres naturals n , x i y , retorni tots els multiconjunts que es poden formar amb $[1..n]$, de manera que cada nombre aparegui entre x i y vegades.
3. Feu una funció `multisets3 :: Int → Int → [[Int]]` que, donats dos naturals n i t , retorni tots els multiconjunts de t nombres que es poden formar amb $[1..n]$.
4. Feu una funció `multisets4 :: Int → Int → Int → Int → [[Int]]` que, donats quatre naturals n, t, x i y , retorni tots els multiconjunts de t nombres que es poden formar amb $[1..n]$ de forma que cada nombre aparegui entre x i y vegades.

Observació

Per tal que no importi l'ordre en que genereu la solució, els jocs de proves ordenen el resultat. Per a això, importeu la funció `sort` del mòdul `Data.List` encara que no la feu servir.

Puntuació

Cada funció puntua 25 punts.

Exemple d'entrada

```
let msort xs = sort (map sort xs)
msort $ multisets1 2 3
msort $ multisets2 2 1 4
msort $ multisets3 3 3
msort $ multisets4 3 1 4 6
```

Exemple de sortida

```
[[], [1], [1,1], [1,1,1], [1,1,1,2], [1,1,1,2,2], [1,1,1,2,2,2], [1,1,2], [1,1,2,2], [1,1,2,2,2], [1,2], [1,1,1,1,2], [1,1,1,1,2,2], [1,1,1,1,2,2,2], [1,1,1,1,2,2,2,2], [1,1,1,2], [1,1,1,2,2], [1,1,1,2,2,2], [1,1,1,1], [1,1,1,2], [1,1,1,3], [1,2,2], [1,2,3], [1,3,3], [2,2,2], [2,2,3], [2,3,3], [3,3,3]]
[[1,1,1,1,2,3], [1,1,1,2,2,3], [1,1,1,2,3,3], [1,1,2,2,2,3], [1,1,2,2,3,3], [1,1,2,3,3,3], [1,2,2,2,2,2,2]]
```

Informació del problema

Autor : Jordi Petit

Generació : 2024-05-02 22:03:49

© Jutge.org, 2006–2024.

<https://jutge.org>