

Discusión de los problemas

17 09 2014

Índice general

1. Problem A: Hardwood Species	1
--------------------------------	---

Capítulo 1

Problem A: Hardwood Species

En este problema hay que determinar, dadas unas especies de árboles, qué porcentaje representan del total. Este tipo de problemas de clasificación y agrupación de datos pueden resultar en *time-outs* para los más novatos en este tipo de competencias sino escogen una estructura correcta para almacenar los datos. De hecho, el competidor tendrá problemas si quiere definir los datos en un simple vector, porque los índices no son enteros $(0, 1, \dots, n)$, por el contrario, los índices están dados por el nombre de la especie, es decir, que necesitamos una estructura para contar las veces que aparece cada especie, por ejemplo:

1	3	2	...	#apariciones
Ash	Gum	Cherry	...	Especie n

Dicho esto, necesitamos una estructura en la que podamos definir el índice como *string*, para lo cual usaremos un *map<string, double> trees*, el *double* es para que al final realicemos la división que nos de el porcentaje sin necesidad de hacer un casting. Dicho porcentaje está dado por la siguiente fórmula:

$$\%especie = \frac{trees[especie]}{total} \quad (1.1)$$

En el ejemplo, *trees[Ash]* valdría 1, *trees[Gum]* valdría 3, y así sucesivamente. Para hallar el total basta con contar cuántos datos nos ingresan.

No sobra resaltar, que por el formato de la entrada se debe utilizar *getline*, de forma que podamos saber cuando termina un caso de prueba, que para este problema está especificado con una línea vacía (que el *cin* ignoraría).

Código fuente

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 // Casos de prueba
6 int c;
7 // Mapa de las especieas
8 map<string,double>trees;
9
10 int main()
11 {
12     while(cin>>c){
13         string l; getline(cin,l);
14         for(int t = 0; t < c; ++t){
15             double pob = 0;
16             // Lee las especies del caso de prueba
17             while(getline(cin,l)){
18                 // Termina con linea vacia
19                 if(l == "")break;
20                 // Cuenta la especie
21                 pob++;
22                 // Si no esta en el mapa lo agrega
23                 if(!trees.count(l))
24                     trees[l]=0;
25                 // Anade +1 a la cuenta de la especie
26                 trees[l]++;
27             }
28             // Recorremos el mapa
29             for(map<string,double>::iterator it = trees.begin
30 (); it != trees.end();++it){
31                 // Imprimimos la especie
32                 cout<<it->first<<" ";
33                 // Imprimimos el % con respecto a la
34                 poblacion
35                 cout<<fixed<<setprecision(4)<<(it->second/pob
36 *100)<<"\n";
37             }
38             trees.clear();
39         }
40     }
41     return 0;
42 }
```