**ZOJ3872DP Beauty of Array**

题意：

{2 *3* 3}求{2} {2 *3*} {2 *3* 3} {*3*} {*3* 3} {3}，剔除集合中的重复元素。和为2+5+5+3+3+3=21。结果即为：dp[0]+dp[1]+dp[2]=2+5+14=21

分析：

设dp[i]表示以元素a[i]结尾的集合总和。如dp[下标2]={2 3 3}+{3 3}+{3}

在数列的最后增加一项时，如i位置项时，如△，情形如下：

{O O O O O △}，等效于{△ O O O O O}。

所以，dp[i]就是如下集合中非重复元素的总和。即：

{△i}

{Oi-1 △i}

…

{O2 … Oi-1 △i}

{O1 … Oi-1 △i}

显然，剔除带有元素△i，剩余的元素都属于dp[i-1]。所以有:

dp[i]=dp[i-1]+ x\*△i.

dp[0]=a[0]

若△i与集合中前面元素都不相同，则有：

dp[i]=dp[i-1]+ i\*△i.

但关键问题是△i可能与前面元素重复，若重复，则不应该计入，因为它已经去构造dp[i-1]了。

问题是如何求得x？显然需要知道前面数列中△i的最后位置。比如加入△i时，△s△t△i元素重复，如下所示，△i的最后位置t。

{O1 …△s …△t …Oi-1 △i}

显然，{Ou…△i} (u>t)，以及{△i}，这些集合中的△i都要计入dp[i]，共有(i-t)个△i。

程序中将用pos[a[i]]表示元素a[i]的最后位置。

下面是运行超时的代码，改用C或C++语言应该可行。

<?php

fscanf(STDIN, "%d", $t);

while ($t--) {

fscanf(STDIN, "%d", $n);

$str = "";

$len = 0;

$a = [];

$dp = [];

$pos = [];

while ($len != $n) {

$str = fgets(STDIN);

$str = str\_replace("\r\n", "", $str);

$tmp = [];

$tmp = explode(' ', $str);

$a = array\_merge($a, $tmp);

$len += count($tmp);

}

for ($i = 0; $i < $n; $i++) {

$a[$i] = (int)$a[$i];

$pos[$a[$i]] = -1;

}

$dp[0] = $a[0];

$pos[$a[0]] = 0;

for ($i = 1; $i < $n; $i++) {

$dp[$i] = $dp[$i - 1] + ($i - $pos[$a[$i]]) \* $a[$i];

$pos[$a[$i]] = $i;

}

$ans = 0;

for ($i = 0; $i < $n; $i++) $ans += $dp[$i];

echo "$ans\n";

}

?>