

[SCOI2008] 着色方案

Description

有 n 个木块排成一行，从左到右依次编号为 $1\sim n$ 。你有 k 种颜色的油漆，其中第 i 种颜色的油漆足够涂 c_i 个木块。所有油漆刚好足够涂满所有木块，即 $c_1 + c_2 + \dots + c_k = n$ 。相邻两个木块涂相同色显得很难看，所以你希望统计任意两个相邻木块颜色不同的着色方案。

Input

第一行为一个正整数 k ，第二行包含 k 个整数 c_1, c_2, \dots, c_k 。

Output

输出一个整数，即方案总数模 $1,000,000,007$ 的结果。

Sample Input

3

1 2 3

Sample Output

10

HINT

50%的数据满足： $1 \leq k \leq 5, 1 \leq c_i \leq 3$ 100%的数据满足： $1 \leq k \leq 15, 1 \leq c_i \leq 5$

题解

这道题看起来像一道dp题。首先看一看数据范围，暴搜是能得50分的。再看100分的数据范围，发现 c_i 最大只有5，是不是感觉发现了米奇妙妙屋？然后我们就可以联想到乌龟棋，是否可以用乌龟棋的方式表示状态？

用 $dp[a][b][c][d][e]$ 表示还能用5次的颜色个数为 a ,还能用4次的颜色个数为 b ,还能用3次的颜色个数为 c ,还能用2次的颜色个数为 d ,还能用1次的颜色个数为 e 时的方案总数，那么我们就可以得到一个状态转移方程（注意这不是最后的方程）：

$$dp[a][b][c][d][e] = dp[a-1][b+1][c][d][e] * a + dp[a][b-1][c+1][d][e] * b \dots + dp[a][b][c][d][e-1] * e$$

不过我们还没有考虑相邻不能重复的情况，那么就再加一维last表示上一次选的颜色在选之前还能用last次，这样每次算的时候如果要选的颜色次数=last-1, -1就可以了。

PS: bzoj还卡空间，把 $dp[17][17][17][17][17]$ 改成 $dp[16][16][16][16][16]$ 才过的。

Code

```
#include<bits/stdc++.h>
using namespace std;

const long long mod=1e9+7;
int k,c[20],tot[6];
long long dp[17][17][17][17][17];

long long dfs(int last,int a,int b,int c,int d,int e){
    long long &ret=dp[last][a][b][c][d][e];
    if(last&&ret)return ret;
    if(a)ret+=dfs(5,a-1,b+1,c,d,e)*a;
    if(b)ret+=dfs(4,a,b-1,c+1,d,e)*(b-(last==5));
    if(c)ret+=dfs(3,a,b,c-1,d+1,e)*(c-(last==4));
    if(d)ret+=dfs(2,a,b,c,d-1,e+1)*(d-(last==3));
    if(e)ret+=dfs(1,a,b,c,d,e-1)*(e-(last==2));
    return ret=ret%mod;
}

int main()
{
    scanf("%d",&k);
    for(int i=1;i<=k;i++)scanf("%d",&c[i]),tot[c[i]]++;
    dp[1][0][0][0][0][0]=1;
    printf("%lld",dfs(0,tot[5],tot[4],tot[3],tot[2],tot[1]));
    return 0;
}
```