AT4531/AtCoder Edu dp J

题意：

n 个数，$1 \leq a_i \leq 3$，每次随机选一个减一，如果是 0 就不管，问全部减成 0 的期望操作次数。

思路：

发现 a_i 的范围仅仅是 3，而每个数又是等效的，所以考虑一个三维 DP

计算期望，考虑从后向前计算，设 $dp[0][0][0]$ 表示末状态的答案，每一维表示 $1,2,3$ 的数量，转移式：

定义：

$$t0 = dp[i][j][k]$$

$$t1 = dp[i-1][j][k]$$

$$t2 = dp[i+1][j-1][k]$$

$$t3 = dp[i][j+1][k-1]$$

$$dp[i][j][k] = (1 + t0 * (n-i-j-k) + t1 * i + t2 * j + t3 * k)/n;$$

看上去好像有环，但是移项搞定，最后的式子：

$$dp[i][j][k] = (1 + t1 * i + t2 * j + t3 * k) * n/(n-i-j-k);$$

顺序有点不好弄，所以记忆化搜索。

其实发现 k 单调，所以 k 为阶段，状态上，顺序枚举 j,i，这样就可以用循环转移了。

参考代码：

```
#include<bits/stdc++.h>
#define y1 y3647
#define earse erase
#define INF 1000000000
#define LL long long
#define pii pair<int,int>
using namespace std;
inline void read(int &x)
{
    x=0;int f=1;
    char ch=getchar();
    while(ch!=45&&(ch>'9'||ch<'0'))ch=getchar();
    if(ch==45){f=-1,ch=getchar();}
    while(ch<='9'&&ch>='0'){x=x*10+ch-48;ch=getchar();}
    x*=f;
}
const int N=305;
int i,j,k,n,s,t,m,s1,s2,s3;
double dp[N][N][N];
double solve(int x1,int x2,int x3)
{
```

```
    if(x1==0&&x2==0&&x3==0)return 0;
    if(dp[x1][x2][x3]>0) return dp[x1][x2][x3];
    if(x1)dp[x1][x2][x3]+=1.0*x1/n*solve(x1-1,x2,x3);
    if(x2)dp[x1][x2][x3]+=1.0*x2/n*solve(x1+1,x2-1,x3);
    if(x3)dp[x1][x2][x3]+=1.0*x3/n*solve(x1,x2+1,x3-1);
    dp[x1][x2][x3]+=1;
    dp[x1][x2][x3]*=1.0*n/(x1+x2+x3);
    return dp[x1][x2][x3];
}
signed main()
{
    //freopen(".in","r",stdin);
    //freopen(".out","w",stdout);
    //freopen(".ans","w",sdtout);
    read(n);
    dp[0][0][0]=0;
    for(i=1;i<=n;i++)
    {
        int x;
        read(x);
        if(x==1)s1++;
        if(x==2)s2++;
        if(x==3)s3++;
    }
    printf("%0.9lf",solve(s1,s2,s3));
    return 0;
}
```

Best Code:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef pair<int,int> P;
typedef pair<int,P> P1;
typedef pair<P,P> P2;
#define pu push
#define pb push_back
#define mp make_pair
#define eps 1e-7
#define INF 1000000000
#define mod 1000000007
#define fi first
#define sc second
#define rep(i,x) for(int i=0;i<x;i++)
#define repn(i,x) for(int i=1;i<=x;i++)
#define SORT(x) sort(x.begin(),x.end())
#define ERASE(x) x.erase(unique(x.begin(),x.end()),x.end())
#define POSL(x,v) (lower_bound(x.begin(),x.end(),v)-x.begin())
#define POSU(x,v) (upper_bound(x.begin(),x.end(),v)-x.begin())
double dp[305][305][305];
int n,a[3];
double rec(int a,int b,int c){
    if(dp[a][b][c] >= -1e16) return dp[a][b][c];
    if(a+b+c == 0) return 0.0;
```

```
    double sum = 0;
    if(a) sum += (double)(a)/(double)(a+b+c)*rec(a-1,b,c);
    if(b) sum += (double)(b)/(double)(a+b+c)*rec(a+1,b-1,c);
    if(c) sum += (double)(c)/(double)(a+b+c)*rec(a,b+1,c-1);
    sum += 1.0*(double)(n)/(double)(a+b+c);
    return dp[a][b][c] = sum;
}
int main(){
    cin>>n;
    for(int i=0;i<305;i++) for(int j=0;j<305;j++) for(int k=0;k<305;k++){
        dp[i][j][k]=-1e18;
    }
    for(int i=0;i<305;i++) for(int j=0;j<305;j++) for(int k=0;k<305;k++){
        if(i+j+k > n) continue;
        dp[i][j][k] = rec(i,j,k);
    }
    rep(i,n){
        int x; cin>>x; a[x-1]++;
    }
    printf("%.12f\n",dp[a[0]][a[1]][a[2]]);
}
//by IH19980412
//https://atcoder.jp/contests/dp/submissions/3943172
//记忆化搜索，无需考虑顺序
```

常见错误：

1.RE：

这道题数据范围虽然是 300 但是涉及到加法的转移，数组应该略微再大一点。

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
using namespace __gnu_pbds;

#define fi first
#define se second
#define mp make_pair
#define pb push_back

typedef long long ll;
typedef pair<ll,ll> ii;
typedef vector<int> vi;
typedef unsigned long long ull;
typedef long double ld;
typedef tree<ii, null_type, less<ii>, rb_tree_tag,
tree_order_statistics_node_update> pbds;

ld dp[305][305][305];
int cnt[3];

bool valid(int x, int y, int z)
{
    return (x>=0&&y>=0&&z>=0);
```

```
}

int main()
{
    ios_base::sync_with_stdio(0); cin.tie(0);
    int n; cin>>n;
    for(int i=0;i<n;i++)
    {
        int x; cin>>x; x--; cnt[x]++;
    }
    dp[0][0][0]=0;
    for(int k=0;k<=cnt[2];k++)
    {
        for(int j=0;j<=n;j++)
        {
            for(int i=0;i<=n;i++)
            {
                if(i==0&&j==0&&k==0) continue;
                dp[i][j][k]=1;
                if(i>0) dp[i][j][k]+=ld(i)/ld(n)*dp[i-1][j][k];
                if(j>0) dp[i][j][k]+=ld(j)/ld(n)*dp[i+1][j-1][k];
                if(k>0) dp[i][j][k]+=ld(k)/ld(n)*dp[i][j+1][k-1];
                dp[i][j][k]*=ld(n)/ld(i+j+k);
                //cerr<<i<<' '<<j<<' '<<k<<' '<<dp[i][j][k]<<'\n';
            }
        }
    }
    cout<<fixed<<setprecision(15)<<dp[cnt[0]][cnt[1]][cnt[2]]<<'\n';
}
//by zscoder
//https://atcoder.jp/contests/dp/submissions/3940324
```

还有一份 ec24 的也是这个错误，就不放了。

2.TLE:

vector 访问多重数组效率低下。

```
#include <bits/stdc++.h>

using namespace std;

int main() {

    int n; cin >> n;

    vector<int> cnt(3, 0);
    for (int i = 0; i < n; ++i) {
        int x; cin >> x;
        cnt[x - 1] += 1;
    }

    vector<vector<vector<double>>> dp(305,
            vector<vector<double>>(305,
                vector<double>(305, 0)));

    for (int k = 0; k <= cnt[2]; ++k) {
```

```
        for (int j = 0; j <= cnt[1] + cnt[2]; ++j) {
            for (int i = 0; i <= cnt[0] + cnt[1] + cnt[2]; ++i) {
                if (i == 0 && j == 0 && k == 0) continue;

                int total = i + j + k;

                dp[i][j][k] = 1;
                if (i > 0) {
                    dp[i][j][k] += 1.0 * i / n * (dp[i - 1][j][k]);
                }
                if (j > 0) {
                    dp[i][j][k] += 1.0 * j / n * (dp[i + 1][j - 1][k]);
                }
                if (k > 0) {
                    dp[i][j][k] += 1.0 * k / n * (dp[i][j + 1][k - 1]);
                }

                dp[i][j][k] /= 1 - 1.0 * (n - total) / n;
            }
        }
    }

    cout << fixed << setprecision(20) << dp[cnt[0]][cnt[1]][cnt[2]] << endl;


    return 0;
}
//by retrograd
//https://atcoder.jp/contests/dp/submissions/3944952
```

使用了低效的 map 实现记忆化

```
#include <bits/stdc++.h>

#define ll          long long
#define pb          push_back
#define pii         pair<int,int>
#define vi          vector<int>
#define vii         vector<pii>
#define mi          map<int,int>
#define mii         map<pii,int>
#define all(a)      (a).begin(),(a).end()
#define x           first
#define y           second
#define sz(x)       (int)x.size()
#define endl        '\n'
#define hell        1000000007
#define rep(i,a,b)  for(int i=a;i<b;i++)
using namespace std;
int n;
map<vi,double> dp;
double sol(vi a){
    if(a[1]==0 and a[2]==0 and a[0]==0) return 0;
    if(dp.count(a)) return dp[a];
    double &ans=dp[a];
    int c0=n-a[0]-a[1]-a[2];
    double cur=0;
```

```
    rep(i,0,3){
        if(a[i]==0) continue;
        a[i]--;
        if(i-1>=0) a[i-1]++;
        cur+=(a[i]+1)*(sol(a)+1);
        if(i-1>=0) a[i-1]--;
        a[i]++;
    }
    return ans=(c0+cur)/(n-c0);
}
void solve(){
    cin>>n;
    vi a(3);
    rep(i,1,n+1){
        int x;
        cin>>x;
        x--;
        a[x]++;
    }
    cout<<fixed<<setprecision(10)<<sol(a)<<endl;
}

signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1;
    // cin>>t;
    while(t--){
        solve();
    }
    return 0;
}
//by hitman623
//https://atcoder.jp/contests/dp/submissions/3941240
```

3.WA

转移的顺序 i,j,k 弄错了

```
#include <bits/stdc++.h>
#define mk make_pair
using namespace std;
typedef long long ll;
const int N = 3e2 + 5;
const int mod = 1e9 + 7;
int n, a[N], s[N];
double dp[N][N][N];
int main() {
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
        s[a[i]]++;
    }
    for (int i = 0; i <= s[1]; ++i) {
        for (int j = 0; j <= s[2]; ++j) {
            for (int k = 0; k <= s[3]; ++k) {
```

```
                if (i + j + k == 0) continue;
                if (i) dp[i][j][k] += (dp[i - 1][j][k] + 1) * i / n;
                if (j) dp[i][j][k] += (dp[i + 1][j - 1][k] + 1) * j / n;
                if (k) dp[i][j][k] += (dp[i][j + 1][k - 1] + 1) * k / n;
                double p = (double)(n - i - j - k) / n;
                dp[i][j][k] += p;
                dp[i][j][k] /= (1 - p);
            }
        }
    }
    printf("%.10f\n", dp[s[1]][s[2]][s[3]]);
}
//by hg_pt
//https://atcoder.jp/contests/dp/submissions/3940770
```