题意：求f(n)=1/1+1/2+1/3+1/4…1/n (1 ≤ n ≤ 108).，精确到10-8

这题让求1/1 + 1/2 + 1/3 + .... + 1/n 的和，然而n有1e8这么大，所以直接打表是会MLE的，所以就用到了一点技巧，每隔50个存一个和，这样200W个数还是可以存的下的。两个之间的数直接加上就行了。

#include<bits/stdc++.h>

#include<vector>

using namespace std;

typedef long long ll;

const int maxn=100000010;//1e8+10

double sum[maxn/50+10];

void init()

{

sum[0]=0;

double t(0);

for(int i=1;i<=maxn;i++)

{

t+=1.0/i;

if(i%50==0)

sum[i/50]=t;

}

}

int main()

{

// freopen("input.txt","r",stdin);

// ios::sync\_with\_stdio(false);

// cin.tie(0);

int T,n;

init();

scanf("%d",&T);

for(int cas=1;cas<=T;cas++)

{

scanf("%d",&n);

int t=n/50;

double ans=sum[t];

for(int i=t\*50+1;i<=n;i++)

ans+=1.0/i;

printf("Case %d: %.8f\n",cas,ans);

}

return 0;

}

解法二：

知识点：

调和级数(即f(n))至今没有一个完全正确的公式，但欧拉给出过一个近似公式：(n很大时)

f(n)≈ln(n)+C+1/2\*n

欧拉常数值：C≈0.57721566490153286060651209

c++ math库中，log即为ln。

#include<bits/stdc++.h>

#include<vector>

using namespace std;

typedef long long ll;

const int maxn=100000010;//1e8+10

double a[10010];

const double c=0.57721566490153286060651209;

int main()

{

// freopen("input.txt","r",stdin);

// ios::sync\_with\_stdio(false);

// cin.tie(0);

int T,n;

scanf("%d",&T);

a[1]=1;

for(int i=2;i<=10000;i++)

a[i]=a[i-1]+1.0/i;

for(int cas=1;cas<=T;cas++)

{

scanf("%d",&n);

if(n<=10000)

{

printf("Case %d: %.8f\n",cas,a[n]);

}

else

{

double x=log(n)+c+1.0/(2\*n);

printf("Case %d: %.8f\n",cas,x);

}

}

return 0;

}