二分法模板

```
double f(double x)
{
    return x * x * x; //以y=x^3为例
}

double get(double y)
{
    double left = 0;
    double right = 1E10;

    //二分100次，使区间足够小
    for (int i = 0; i < 100; i++)
    {
        double mid = (left + right) / 2;

        if (f(mid) < y)
            left = mid;
        else
            right = mid;
    }

    return left;
}
```

```
int binarySearch(std::vector<int>&num,int left,int right,int target)
{
    if(left>right)
        return -1;
    int mid=left+(right-left)/2;
    if(nums[mid]==target)
    {
        reutrn mid;
    }
    else if(nums[mid]<target)
    {
        return binarySearch(nums,mid+1,right,target)
    }
    else
        return binarySearch(nums,left,mid-1,target);
}

int binarySearch(std::vector<int>&nums,int target)
{
    if(nums.size()==0)
        return -1;
    int left=0;
    int right=nums.size()-1;
    while(left<=right)
    {
```

```
        int mid=left+(right-left)/2;
        if(nums[mid]==target)
        {
            return mid;
        }
        else if(nums[mid]<target)
        {
            left=mid+1;
        }
        else
        {
            right=mid-1;
        }
    }
    return -1;
}


//二分法假定解

double f(double x)
{
    return x*x*x;
}
double get(double y)
{
    double left=0;
    double right=1e10;
    for(int i=0;i<100;i++)  //思路2：while (right - left > 精度) { ... }
    {
        double mid=(left+right)/2;
        if(f(mid)<y)
            left=mid;
        else
            right=mid;
    }
    return left;
}
```