Efficient algorithm for calculating the area of spherical compound area

Hangzhou Xuejun Middle School Zhou Renfei

Pick want

On the spherical surface, repeated Boolean operations such as intersection, union, and difference of the spherical crown-shaped area can express many complex areas, which are called "composite areas" in this article. Calculating the area of the spherical composite area is an important problem, and it has a wide range of applications in reality. This article proposes a definition of the area of a complex area in the form of a Boolean expression, and proposes an excellent algorithm to $O(nm \log m)$ The worst-case running time to find the exact value of the area of the spherical composite region, where n Is the number of spherical crown-shaped areas,m Is the length of the boolean expression.

1 introduction

On a given sphere n Spherical crown $D_1, D_2, \ldots, D_n$, And a length of m Boolean expression of to define a compound area D^*. Among them, these spherical caps are two different; Boolean expressions are given explicitly, and each area can appear multiple times in the expression. How to calculate this composite area D^* The area of ?? is the main issue discussed in this article-the area of the spherical compound area.

This type of problem has a wide range of applications in reality, such as calculating the amount of raw materials for spherical crafts, calculating the area of a specific geographic area, calculating the coverage of communication base stations, calculating the effective service range of the "Beidou" satellite positioning system on the surface of the earth, etc. Wait. These applications are particularly important in computer-aided design and geographic information systems.

The special case of this kind of problem on the plane has appeared in the algorithm competition for a long time. already NOI 2004, The question of "Rainfall" appeared, requiring to calculate the area of the union of several parallelograms on the plane. In later years,NOI There have been several times of "area union" problems on the plane, such as the more famous NOI 2005"Lemon Tree Under the Moon"[2]. As for the area of the spherical compound area, ICPC Appeared once in the event[6].

Previously, there was no universal method to solve such problems. There is a traditional numerical integration method that can calculate an approximate value of the area of the composite region. In addition, there is also a scan line method, by cutting the composite area, in $O(n^2 m \log m)$ Calculate the exact value of the area of the planar composite area within the worst time complexity. This article proposes an excellent algorithm, in $O(nm \log m)$ Calculate the accurate value of the area of the composite area of the plane or the sphere within the worst time complexity, which has the advantages of high efficiency, high precision, good numerical stability, and strong scalability.

This article consists of six chapters. The second chapter introduces a simpler problem-the problem of flat area union-and proposes the Green formula method to obtain an upper bound of the running time as $O(n^2 \log n)$ The precise algorithm. The third chapter applies the Green formula method to the spherical surface, and solves the spherical area union problem with the same upper bound of the running time. Chapter 4 uses Boolean operators and Boolean expressions to put forward a precise definition of the area of the spherical compound area and introduces an excellent algorithm to solve the problem. Chapter 5 discusses some application scenarios of this algorithm and shows how to make appropriate adjustments to the various stages of the algorithm to adapt to specific problems. The sixth chapter is a brief summary.

2 Flat area and problem

problem 1(Flat area and problem) On a given plane n Circular area $D_1, \ldots, D_n$,definition

$$D^* = D_1 \cup D_2 \cup \ldots \cup D_n,$$

Each of the circular areas D_i Refers to the inside of a circle. Calculation D^* Area.

This chapter proposes an algorithm based on Green's formula to solve the above problems.

Green formula Assume $\Omega \subset \mathbb{R}^2$ It is a closed area enclosed by a finite piece of smooth curve. If function $P(x, y)$ with $Q(x, y)$ in Ω Upper continuous, and continuous partial derivative $\partial Q/\partial x$ with $\partial P/\partial y$, Then there is

$$\oint_{\partial\Omega} P\mathrm{d}x + Q\mathrm{d}y = \iint_{\Omega} \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) \mathrm{d}x\,\mathrm{d}y,$$

among them $\partial\Omega$ Is the area Ω Borders. Its orientation is determined like this: a person follows $\partial\Omega$ When traveling in the positive direction, the area Ω Always on the left of this person.

The proof of the theorem can be found in the mathematical analysis textbook. The proof is no longer described here, but how to use the theorem to calculate D^* Area.

Remember S_{D^*} Means D^* Area, then

$$S_{D^*} = \iint_{D^*} \mathrm{d}x\,\mathrm{d}y.$$

If order

$$\begin{cases} \Omega = D^*, \\ P(x,y) = -y, \\ Q(x,y) = 0, \end{cases}$$

then Ω, P, Q Meet the application conditions of Green's formula. Substituting into Green?s formula, we get

$$\begin{aligned} \oint_{\partial D^*} (-y)\,\mathrm{d}x &= \iint_{D^*} \mathrm{d}x\,\mathrm{d}y \\ &= S_{D^*}, \end{aligned}$$

So as to convert the calculated area into a calculated boundary ∂D^* The curve integral.

Figure 1 Is the problem 1 An example of where D^* Is painted in lavender, the circle is marked with a dotted line,∂D^* It is marked with a solid line. as the picture shows,∂D^* Is composed of several arcs, let these arcs be $A_1, \ldots, A_m$, According to the additivity of points, there are

$$\oint_{\partial D^*} (-y)\,\mathrm{d}x = \sum_{i=1}^{m} \int_{A_i} (-y)\,\mathrm{d}x,$$

Each of them A_i The directions are all counterclockwise. We use the right side of the above formula to calculate the left side of the above formula. So the question 1 Is transformed into two sub-problems:

problem 1.1 Find out $A_1, A_2, \ldots, A_m$.

problem 1.2 Calculation $\int_{A_i} (-y)\,\mathrm{d}x$.

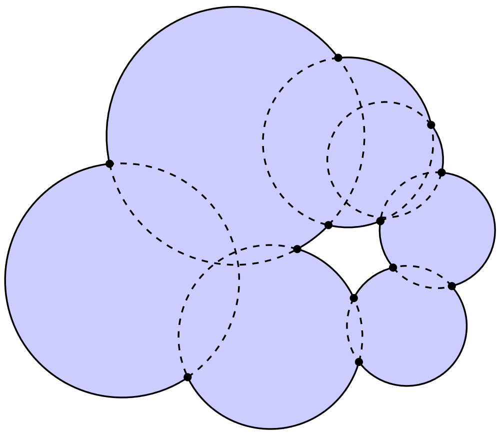


Figure 1

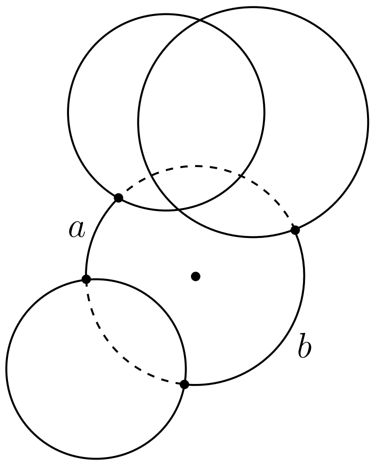


Figure 2

Discuss the problem first 1.1. Without loss of generality, suppose the circle $D_1, D_2, \ldots, D_n$ The two are different; otherwise, for circles that appear multiple times, we can keep one and delete the others without changing the answer to the question. As shown 2 As shown, the part of each circle that is not covered by other circles (take the center circle as an example: arc a, b)constitute ∂D^*; The part covered by other circles (the dotted line) is not ∂D^* Part. Find the composition ∂D^* For each circle, we are required to calculate the part of the circle that is not covered by other circles.

Define the "polar angle" operation. Suppose the center of the circle we are dealing with is O; Define a point on the circle P Polar angle $\mathsf{Angle}(P)$ As: from x Axis positive direction to vector $\overrightarrow{OP}$ The angle the direction has been turned. This is a directional angle. Turn counterclockwise to take the positive sign, and the value range is specified as $[-\pi, \pi)$. The polar angle calculation is from the circle to $[-\pi, \pi)$ One-to-one mapping. We turn to calculations $[-\pi, \pi)$ On the uncovered part, each circle in the circle O The covered part (if any) is a circular arc, and its corresponding polar angle is $[-\pi, \pi)$ Up to the union of two intervals. The problem is further transformed into:

problem 1.1a given $[-\pi, \pi)$ Up n' An interval, find the part that is not covered by any interval, and express it with a number of disjoint intervals. among them $n' \leq 2n$.

This is a classic problem, here is a brief introduction to the upper bound of the running time as $O(n' \log n')$ Algorithm.

Use the idea of scanning lines to make moving points P From $-\pi$ Gradually move to π, In this process, there are several intervals to maintain P. This value is from 1 become 0, And from 0 become 1 The position is the left and right end points of the uncovered interval. Before this process, all interval endpoints need to be sorted, and the upper bound of the running time is $O(n' \log n')$.

Find out $[-\pi, \pi)$ After several intervals that are not covered on the above, the arcs corresponding to them can be calculated in linear time. for n Perform the above calculations for each circle, just in $O(n^2 \log n)$ Found within the worst time complexity of $A_1, \ldots, A_m$,solved the problem 1.1.

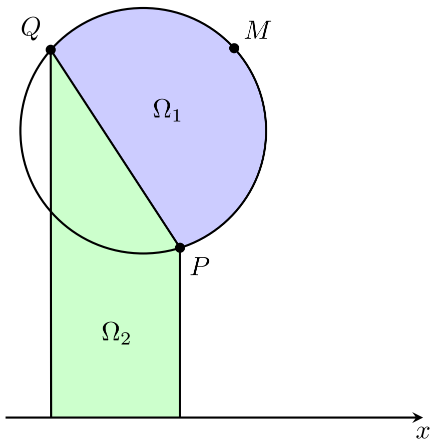


Figure 3

Finally, discuss the problem 1.2. As shown 3 As shown, suppose we want a circular arc $A_i = \widehat{PMQ}$ Calculation $\int_{A_i} (-y)\,\mathrm{d}x$(The direction of the curve is counterclockwise,$P \to Q$). area Ω_1 By A_i with $\overrightarrow{QP}$ The enclosed arcuate area is marked in light blue in the figure;Ω_2 By $\overrightarrow{QP}$,x The area enclosed by the axis and two vertical lines. To the area Ω_1 Applying Green?s formula, we get

$$\begin{aligned}
S_{\Omega_1} &= \oint_{\partial\Omega_1} (-y)\,\mathrm{d}x \\
&= \int_{A_i} (-y)\,\mathrm{d}x + \int_{\overrightarrow{QP}} (-y)\,\mathrm{d}x \\
\Rightarrow \quad \int_{A_i} (-y)\,\mathrm{d}x &= S_{\Omega_1} + \int_{\overrightarrow{QP}} y\,\mathrm{d}x \\
&= S_{\Omega_1} + S_{\Omega_2},
\end{aligned}$$

among them S_{Ω_2} Is the area Ω_2 The directed area: if Q The abscissa is greater than P,then $S_{\Omega_2} < 0$. The last step of derivation is used $\overrightarrow{QP}$ The geometric meaning of the integral.

Ω_1 Is a bow shape,Ω_2 They are trapezoidal, and their area is easy to calculate. Using the above formula, a circular arc can be calculated in a constant time A_i Points to solve the problem 1.2.

Finally, we re-examine the problem 1. we are at $O(n^2 \log n)$ Within the worst time complexity, find the composition ∂D^* All arcs of $A_1, \ldots, A_m$,among them m There is an upper bound n^2. Then we give each A_i Calculate its curve integrals, each takes a constant time, and add them to get S_{D^*}. Thus, we get an upper bound on running time as $O(n^2 \log n)$ Excellent algorithm that solves the problem 1.

3 Spherical area union problem

In the previous chapter, we used Green?s formula to D^* The double integral on is transformed into ∂D^* On the curve integral, and then the composition ∂D^* Calculate the curve integrals of each arc separately. In this chapter, we use the same idea to solve the problem of spherical area union.

problem 2(Spherical area and problem) On a given sphere n Spherical crown $D_1, \ldots, D_n$, The crown cannot be larger than the hemisphere. definition

$$D^* = D_1 \cup D_2 \cup \ldots \cup D_n,$$

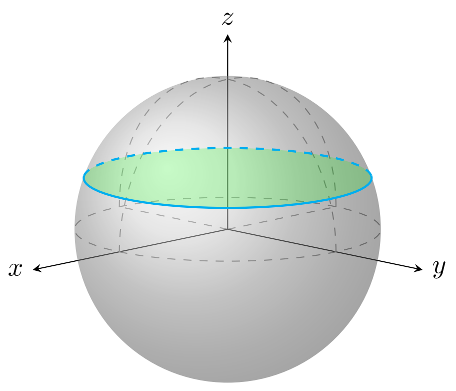
Calculation D^* Area.

Here we stipulate that the crown cannot be larger than the hemisphere. It does not mean that the large crown cannot be processed, but because after the Boolean operation is proposed, we can use the complement of the smaller crown to represent the crown that exceeds the hemisphere, so there is no need Discuss this situation. Making this provision can provide convenience for geometric processing.

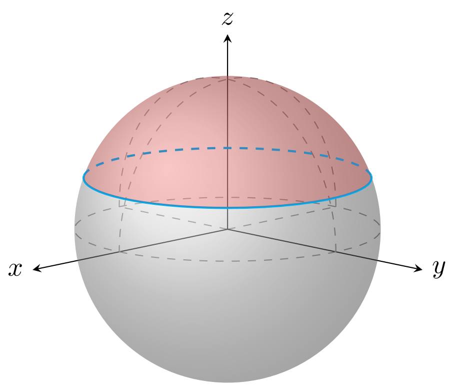
Before introducing the solutions to the above problems, let us briefly introduce the basic knowledge of spherical geometry and agree on the meaning of some terms.

(1) We assume that the sphere in the problem is a unit sphere, that is, the center of the sphere is at the origin $O = (0, 0, 0)$With a radius of 1. This is possible because the impact of the translation and zooming of the ball on the answer to the question is easy to deal with.

(2) Use a plane to cut the sphere, the cut part is a**round**(As shown 4a Shown). The part of the sphere on one side of the plane is called**Ball crown**(As shown 4b Shown). There is a corresponding relationship between the two, the circle is the boundary of the spherical cap, and the spherical cap is the inside of the circle. Where there is no need to distinguish between them, their names can be mixed.



(a) round



(b) Ball crown

Figure 4

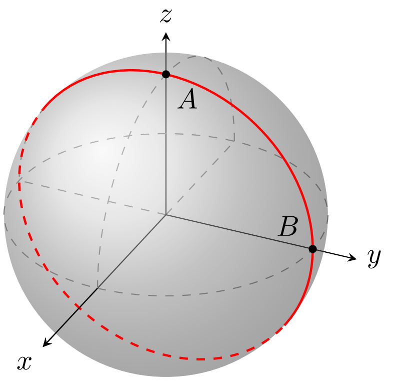
For a ball crown, we call it**vertex**Is its center point on the sphere. It is on the crown, and the distance from each point on the border of the crown is equal. Call it**Underside**Is the plane of the corresponding circle.

(3) We call the two points on the sphere**Opposite**Yes, if and only if their connection passes through the center of the sphere. This is equivalent to their symmetry about the center of the sphere.

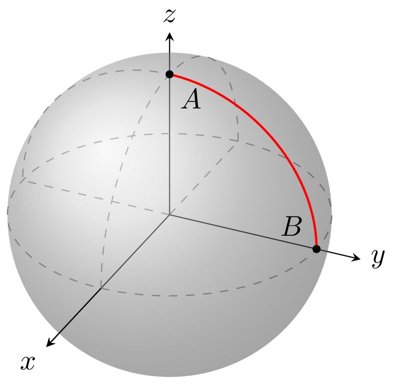
(4) **Big circle**It is the largest circle, that is, the circle intercepted by the plane of the center of the sphere to intercept the sphere. Except for the great circle, all other circles are called**Madoka**. The great circle divides the entire sphere into two equal parts, namely two hemispheres. The position of a great circle in spherical geometry is like a straight line in plane geometry. For two different points on the sphere A, B, If they do not align with each other, there will be a unique great circle passing through them at the same time (as shown in the figure 5a Shown), called**Spherical straight line** AB Or big circle AB. at this time A, B Divide this great circle into two segments with unequal lengths, and the shorter segment is called**Spherical line segment** AB(As shown 5b Shown), it is from the spherical surface A To B The shortest route.

Each big circle AB Uniquely determine a plane OAB.

in case A, B On the opposite diameter, the spherical surface is from A To B There are unlimited shortest routes. Any one ever A, B The great circle,A, B Divide it into two equal lengths, each of which starts from A To B The shortest route. We call any one of the shortest paths as the generalized spherical segment between them. But in order to specify a generalized spherical segment, we need to give the division $\boldsymbol{A}, \boldsymbol{B}$ Besides, another point on it $\boldsymbol{M}$.



(a) Spherical straight line



(b) Spherical line segment

Figure 5

(5) in case AB, AC Are all spherical segments, so you can define**Spherical angle** $\angle BAC$. As shown 6 Shown, over A Point for $\overset{\frown}{AB}$ Tangent AF, Make $\overset{\frown}{AC}$ Tangent AE,definition $\angle BAC = \angle FAE$. This is equivalent to a plane $\boldsymbol{OAB}$ versus $\boldsymbol{OAC}$ The dihedral angle between.

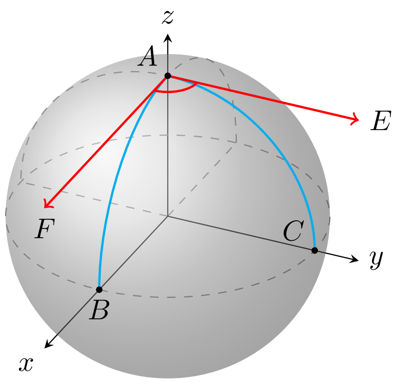


Figure 6

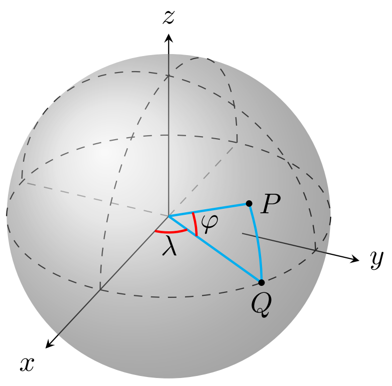


Figure 7

(6) If there are three points on the sphere A, B, C It can be defined if they are different, have different diameters, and are not on the same great circle.**Spherical triangle** ΔABC. Its three sides are spherical segments AB, BC, CA. These three sides divide the entire sphere into two parts, the smaller part is ΔABC internal. The three internal angles of the spherical triangle are all smaller than the straight angle.

(7) As in the case on Earth, the definition**South Pole** $P_S = (0, 0, -1)$,**North pole** $P_N = (0, 0, 1)$, And define the warp and weft. Specifically, connect P_N versus P_S All spherical segments of are called**warp**;flat $z = z_0$ The circle obtained by cutting the sphere is called**weft**.

We can use latitude and longitude to represent points on the sphere, as shown in the figure 7 Shown. For each point on the sphere $P = (x, y, z)$, Let the real number $\lambda \in [0, 2\pi)$ with $\varphi \in [-\pi/2, \pi/2]$ Satisfy

$$\begin{cases} x = \cos\varphi \cdot \cos\lambda, \\ y = \cos\varphi \cdot \sin\lambda, \\ z = \sin\varphi, \end{cases}$$

Just called (λ, φ) Yes point P The latitude and longitude said. among them λ Called longitude,φ Called latitude.

There are multiple ways of expressing the two poles in latitude and longitude. We delete the two poles from the sphere to form**Depolarization**, This will not change the area we want to calculate. Every point on the polar sphere P Have a unique latitude and longitude representation

$$(\lambda, \varphi) \in [0, 2\pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

The right side of the above formula is a rectangular area on the plane, which is called**Latitude and longitude plane**. On the latitude and longitude plane, stipulate λ Is the abscissa,φ Is the ordinate.

The longitude and latitude representation of a point establishes a one-to-one mapping from the depolarizing sphere to the longitude and latitude plane. We can use this mapping to convert the area of the area on the sphere into a double integral on the latitude and longitude plane, and then link it with the curve integral through Green's formula.

theorem 1 Hypothesis Ω It is an area surrounded by a finite piece of smooth curve on the sphere, and the north pole is P_N Not in Ω On the boundary, then there are

$$S_\Omega = \oint_{\partial\Omega} (-\sin\varphi - 1)\,\mathrm{d}\lambda + [P_N \in \Omega]\, 4\pi.$$

prove

$$S_\Omega = \iint_\Omega \mathrm{d}a, \tag{1}$$

Where the right side of the above formula is right Ω Surface integral,$\mathrm{d}a$ Is the area micro-element. As shown 8 As shown, there are

$$\mathrm{d}a = \cos\varphi \cdot \mathrm{d}\lambda \cdot \mathrm{d}\varphi.$$

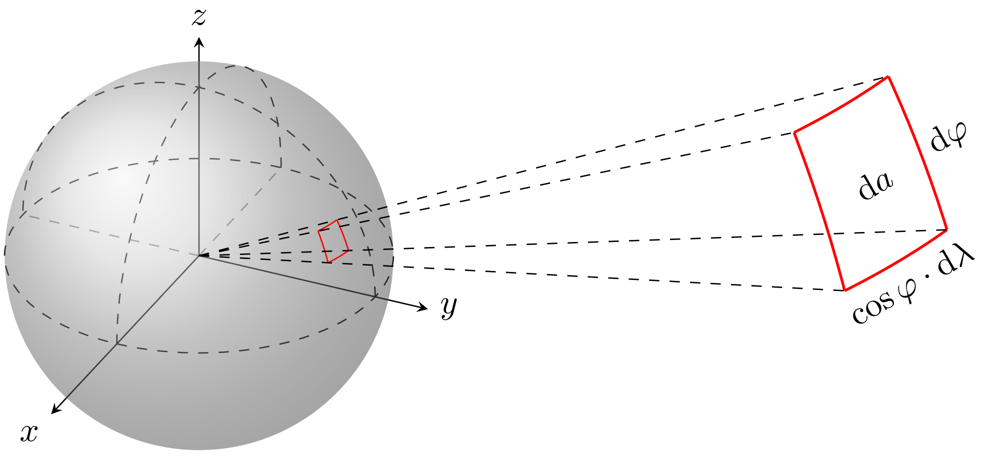


Figure 8

Hypothesis $\hat{\Omega}$ Yes Ω The corresponding area on the latitude and longitude plane.(1) The formula can be rewritten as

$$S_\Omega = \iint_{\hat{\Omega}} \cos\varphi \,\mathrm{d}\lambda \,\mathrm{d}\varphi,$$

On the right is the double integral on the latitude and longitude plane. Please note that the integral on the right side of the above formula does not contain the poles. The poles are removed in this step, which will not change the area of the region.

According to Green?s formula, there are

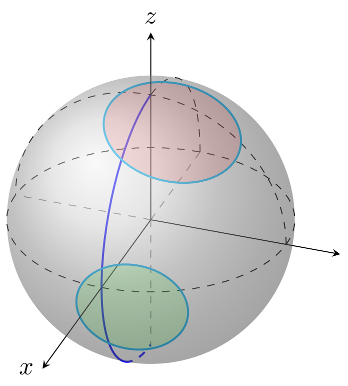
$$\oint_{\partial\hat{\Omega}} P\mathrm{d}\lambda + Q\mathrm{d}\varphi = \iint_{\hat{\Omega}} \left(\frac{\partial Q}{\partial \lambda} - \frac{\partial P}{\partial \varphi}\right) \mathrm{d}\lambda \,\mathrm{d}\varphi \tag{2}$$

Established. make

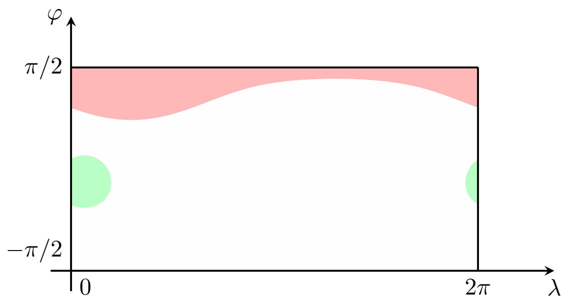
$$\begin{cases} P(\lambda,\varphi) = -\sin\varphi - 1, \\ Q(\lambda,\varphi) = 0, \end{cases}$$

And substitute (2),get

$$\begin{aligned}\oint_{\partial\hat{\Omega}} (-\sin\varphi - 1)\,\mathrm{d}\lambda &= \iint_{\hat{\Omega}} \cos\varphi \,\mathrm{d}\lambda \,\mathrm{d}\varphi \\ &= S_\Omega.\end{aligned}$$



(a) Depolarization



(b) Latitude and longitude plane

Figure 9

$\hat{\Omega}$ The boundary can be divided into two parts: the first is at the boundary of the latitude and longitude plane, and the second is inside the latitude and longitude plane. As shown 9 As shown in the second part of $\partial\Omega$ the same. In the first part, there are $\mathrm{d}\lambda = 0$, For the lower boundary $-\sin\varphi - 1 = 0$, So the curve integrals along these three boundaries are all 0, So we don?t need to care about which parts are $\hat{\Omega}$ Borders. If the north pole is at Ω , Then the entire upper boundary of the latitude and longitude plane is $\hat{\Omega}$ The boundary of, the direction is from right to left, and its integral value is

$$\begin{aligned}&\int_{2\pi}^{0}\left(-\sin\frac{\pi}{2}-1\right)\mathrm{d}\lambda\\ =&\int_{2\pi}^{0}(-2)\,\mathrm{d}\lambda\\ =&\ 4\pi.\end{aligned}$$

Conversely, if the North Pole is not Ω , Then the entire upper boundary is not $\hat{\Omega}$ Borders. In summary, for $\partial\hat{\Omega}$ The first part, there are

$$\int_{\text{Part-1}}(-\sin\varphi-1)\,\mathrm{d}\lambda = [P_N\in\Omega]\,4\pi.$$

Combine Part-2 = $\partial\Omega$,get

$$\begin{aligned}\oint_{\partial\hat{\Omega}}(-\sin\varphi-1)\,\mathrm{d}\lambda &= \int_{\text{Part-1}}(-\sin\varphi-1)\,\mathrm{d}\lambda+\int_{\text{Part-2}}(-\sin\varphi-1)\,\mathrm{d}\lambda\\ &= [P_N\in\Omega]\,4\pi+\oint_{\partial\Omega}(-\sin\varphi-1)\,\mathrm{d}\lambda.\end{aligned}$$

The left side of the equation is equal to S_{Ω}, The right side and the theorem 1 it's the same. □

Please pay attention to the relationship between the depolarizing sphere and the latitude and longitude plane in the above proof process. The integral we perform on the sphere contains only variables λ, φ, Which is the same as the latitude and longitude plane. For points, curves, and regions, there is no difference between them on the sphere and the latitude and longitude planes, and their integral values are also the same in the two forms. The only difference is that in the process of applying Green?s formula on the latitude and longitude plane, when the boundary is defined by the area $\partial\Omega$ versus $\partial\hat{\Omega}$ The difference-the latter is partly more than the former, which is described above $\partial\hat{\Omega}$ The first part.

Another point that needs to be added is that when the curve crosses the meridian,λ There is a discontinuity. At this time, we can cut the curve into two sections at the position of the discontinuity point. λ They are all continuous, so their integrals are well defined; using the additivity of integrals, the integral of the entire curve is defined as the sum of these two integrals. For narrative convenience, the general notation of curve integral is still used in the text to express it.

inference S_Ω versus $\oint_{\partial\Omega} (-\sin\varphi - 1)\,\mathrm{d}\lambda$ They can be deduced from each other, that is, as long as one of them is known, the value of the other can be calculated in a constant time.

Back to the question 2. We assume $D_1, \ldots, D_n$ Pairs are different.

Before solving this problem, rotate the entire sphere at a random angle. Specifically, a point is selected uniformly and randomly on the sphere P'_N, And rotate the point to $(0, 0, 1)$. After the rotation is completed, define the north and south poles, latitude and longitude according to the previous method. Which makes $D_1, \ldots, D_n$ The probability of a circle passing through the North Pole is 0, We assume it will not happen. In addition, for an arc, if its position is determined by $D_1, \ldots, D_n$ OK, and has nothing to do with the selection of the rotation point (for example, the connection between two spherical crown vertices), then we also assume that it does not pass through the North Pole.

According to inference, the problem 2 Turn into calculation

$$\oint_{\partial D^*} (-\sin\varphi - 1)\,\mathrm{d}\lambda.$$

Same as the previous chapter,∂D^* It is made up of arcs?that is, the parts of each circle that are not covered by other circles. So the problem is transformed into two sub-problems:

problem 2.1 Find the arcs of each circle that are not covered by other circles.

problem 2.2 For a certain section of directed arc C, Calculate its curve integral $\int_C (-\sin\varphi - 1)\,\mathrm{d}\lambda$.

Discuss the problem first 2.1. Suppose we want to O' Find the uncovered part of the circle. The circle is on the plane α Up and click O' Is a plane α The center of the circle.

In plane α Create Cartesian Coordinate System on $xO'y$,and O' Is the center to define the polar angle operation, and the point on the circle is the same as $[-\pi, \pi)$ Establish a one-to-one mapping between. Use the problem solved in the previous chapter 1.1 Method, you can find the uncovered part of the circle, they are used as $\partial\Omega$ The orientation of is also easy to determine, so I won?t repeat it here.

Discuss the problem again 2.2. As shown 10a As shown, suppose we want to calculate the arc $C_1 = \overset{\frown}{PQ}$ Curve integral

$$\int_{C_1} (-\sin\varphi - 1)\,\mathrm{d}\lambda,$$

Where the direction of the curve is $P \to Q$.

The first situation:C_1 It's part of Madoka. Directed spherical line segment $C_2 = \overrightarrow{QP}$,then C_1 versus C_2 Surrounded by an arcuate area A. According to inference, it can be derived from S_A roll out

$$\int_{C_1} (-\sin\varphi - 1)\,\mathrm{d}\lambda + \int_{C_2} (-\sin\varphi - 1)\,\mathrm{d}\lambda,$$

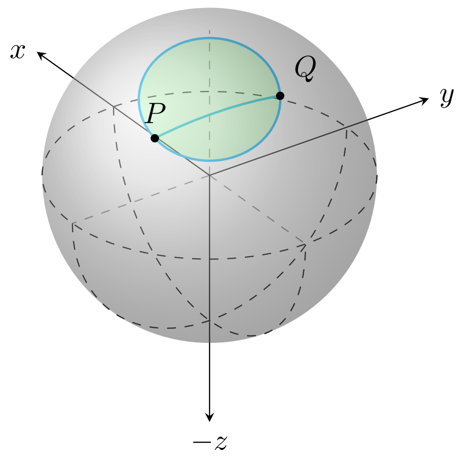
Rejoin C_2 Points to launch C_1 Of points. The problem is transformed into two parts:

problem 2.2a Calculate the area of the bow on the sphere.

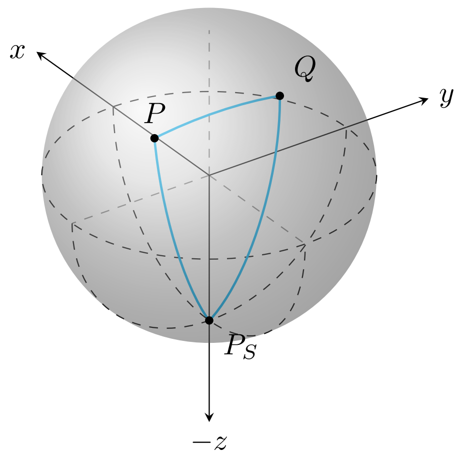
problem 2.2b Given a directed sphere segment $C_2 = \overrightarrow{QP}$, Calculate

$$\int_{C_2} (-\sin\varphi - 1)\,\mathrm{d}\lambda.$$

The second case:C_1 It is part of the great circle. in case C_1 Greater than or equal to half a great circle, then take C_1 Midpoint M, Divide it into two spherical segments $\overrightarrow{QM}$ with $\overrightarrow{MP}$;otherwise,C_1 It is a spherical segment in itself. Which turns into a problem 2.2b.



(a)



(b)

Figure 10

Discuss the problem first 2.2a. By connecting the two end points of the bow shape and the apex of the spherical cap where it is located, we express the bow shape as the sum or difference of a sector and a triangle. The problem is translated into the following two:

problem 2.2c Calculate the area of the spherical sector.

problem 2.2d Calculate the area of the spherical triangle.

Then discuss the problem 2.2b. connection $P_S P$ with $P_S Q$And judge $\Delta P_S PQ$ in $\overrightarrow{QP}$ Which side. If it is on the right, as shown 10b Shown, then calculate $\overrightarrow{PQ}$ The integral of, then take the opposite number. According to the inference, there are

$$\begin{aligned} S_{\Delta P_S PQ} &= \oint_{\partial \Delta P_S PQ} (-\sin\varphi - 1)\,\mathrm{d}\lambda \\ &= \int_{\overrightarrow{P_S P}} (-\sin\varphi - 1)\,\mathrm{d}\lambda + \int_{\overrightarrow{PQ}} (-\sin\varphi - 1)\,\mathrm{d}\lambda + \int_{\overrightarrow{QP_S}} (-\sin\varphi - 1)\,\mathrm{d}\lambda \\ &= \int_{\overrightarrow{PQ}} (-\sin\varphi - 1)\,\mathrm{d}\lambda. \end{aligned}$$

The last step is because for $\overrightarrow{P_S P}$ with $\overrightarrow{QP_S}$ In terms of$\mathrm{d}\lambda = 0$, So its integral value is 0.

So far the problem 2.2b change into 2.2d.

Get the area formula of the spherical crown by simple calculation $S = 2\pi h$,among them h Is the distance from the top of the crown to the bottom. Using the symmetry of the spherical crown, we can easily derive the area formula of the sector

$$S = 2\pi h\theta,$$

among them θ Is the angle of the sector.

Regarding the area of a spherical triangle,[4] Point out

$$S_{\Delta ABC} = \angle A + \angle B + \angle C - \pi.$$

among them $\angle A, \angle B, \angle C$ Respectively expressed as A, B, C Is the spherical angle of the vertex.[7] Gives a formula with better numerical stability

$$\tan \frac{S_{\Delta ABC}}{2} = \frac{\tan\frac{1}{2}a \tan\frac{1}{2}b \sin\angle C}{1 + \tan\frac{1}{2}a \tan\frac{1}{2}b \cos\angle C},$$

among them a,b Respectively refer to $\angle A$,$\angle B$ Opposite side BC versus CA length.

So far, the problem 2 Has been completely resolved. The general structure of the algorithm and the problem solved in the previous chapter 1 The method is similar, and has the same upper bound on running time $O(n^2 \log n)$.

4 Boolean operation

In real applications, what we want to calculate is usually not n Combination of regions. For example, in the space around the earth there are n Positioning satellites, hoping to find out how large an area on the ground can be directly connected to at least three satellites. For another example, when assisting in the design of handicrafts, the user repeatedly uses operations such as intersection, union, and complement to artificially specify a complex area. Even if only the area union problem is considered, the intersection of three hemispheres can be used to represent a spherical triangle, and then the union of triangles can be used to represent a spherical polygon. This raises the requirement to calculate the area of a more general composite area.

k Meta-boolean function $F(x_1, \dots, x_k)$ Is an $\{0,1\}^k \mapsto \{0,1\}$ Mapping.$x_1, \dots, x_k$ Called its parameters.

Hypothesis $D_1, \dots, D_n$ Is the given crown. Can use one n Meta-boolean function $F(\lambda_1, \dots, \lambda_n)$ To define the composite area D^*:

$$D^* = \{P \mid F(\lambda_1, \lambda_2, \dots, \lambda_n) = 1\},$$

which is

$$[P \in D^*] = F(\lambda_1, \lambda_2, \dots, \lambda_n) \qquad (\forall P),$$

among them λ_i Representation point P Is it in D_i Middle, that is $\lambda_i = [P \in D_i]$. Strictly speaking,λ_i Is about P The function $\lambda_i(P)$, But for clarity, at the point we are investigating P Under the very clear premise, its parameters are omitted P, This omission will be used below. These two formulas indicate the Boolean function F Defined $D_1, \dots, D_n$ Which combination of the in the composite area D^* in.

There is a class of basic and simple Boolean functions called Boolean operators. For a Boolean operator $f(x_1, \dots, x_k)$, There must be a data structure that supports:

(1) INITIALIZE. Initialize the data structure, let $x_i = 0$ $(\forall i \in [1, k])$.

(2) MODIFY(i, b). Modify the value of the parameter, so $x_i = b$.

(3) QUERY. Query the value of the function, namely $f(x_1, \dots, x_k)$.

among them INITIALIZE Operation needs to be in $O(k)$ Completed within the time complexity of MODIFY with QUERY It must be completed within a constant running time.

We treat the internal logic of Boolean operators as a black box, and obtain information only through its data structure. all $\Theta(1)$ Meta, computable Boolean functions are all Boolean operators.

We define Boolean expressions recursively. A boolean expression α Or some λ_i ($i \in [1, n]$), Or have

$$\alpha = f(\beta_1, \beta_2, \ldots, \beta_k),$$

among them f Is a Boolean operator,$\beta_1 \ldots, \beta_k$ Each is a Boolean expression.

Definition expression α length

$$|\alpha| = \begin{cases} 1, & \text{if } \alpha = \lambda_i, \\ 1 + \sum_{j=1}^{k} |\beta_j|, & \text{if } \alpha = f(\beta_1, \ldots, \beta_k). \end{cases}$$

We give Boolean functions in the form of Boolean expressions F, Formally raise the following questions:

problem 3(Spherical compound area area problem) Given sphere n Spherical crown $D_1, \ldots, D_n$. Given another length m Boolean expression α_F,definition

$$D^* = \{P \mid F(\lambda_1, \lambda_2, \ldots, \lambda_n) = 1\},$$

among them $F(\lambda_1, \ldots, \lambda_n) = \alpha_F$ Boolean expression α_F Defined n Meta Boolean function. Hope to calculate D^* Area.

Still assume $D_1, \ldots, D_n$ The two are different, and no crown is larger than the hemisphere. This assumption can be made because a spherical cap larger than a hemisphere can be represented by the complement of a smaller spherical cap. Furthermore, we assume that their boundaries do not coincide with each other, that is, there are no two spherical caps. D_i with D_j, Their sum is equal to the entire sphere.

Using the method in the previous chapter, we only need to ∂D^* Can calculate D^* The area of each circle ∂D_i.

In the previous chapter,

$$F = \lambda_1 \vee \lambda_2 \vee \ldots \vee \lambda_n,$$

We judged that a circular arc is ∂D^* In, if and only if it is not covered by other crowns, and the direction is counterclockwise relative to the inside of the crown. This is because, for a directed arc that meets the conditions, its left side (that is, the inner side relative to the spherical cap) is at D^* Inside, while the right side is D^* Outside, so it is ∂D^* For the arc covered by other spherical caps, its left and right sides are in D^* Inside, so it's not ∂D^* Ingredients. Following this idea, if A Yes ∂D_i A directed arc of$A$ On the left D^* Inside, on the right D^* Outside, then $A \subset \partial D^*$;in case A On the left D^* Outside, on the right D^* Inside, then A The reverse is ∂D^* Ingredients. To this end, we calculate the following questions:

problem 3a Calculation ∂D_i All parts of the D^* Inside.

problem 3b Calculation ∂D_i All parts of the D^* Inside.

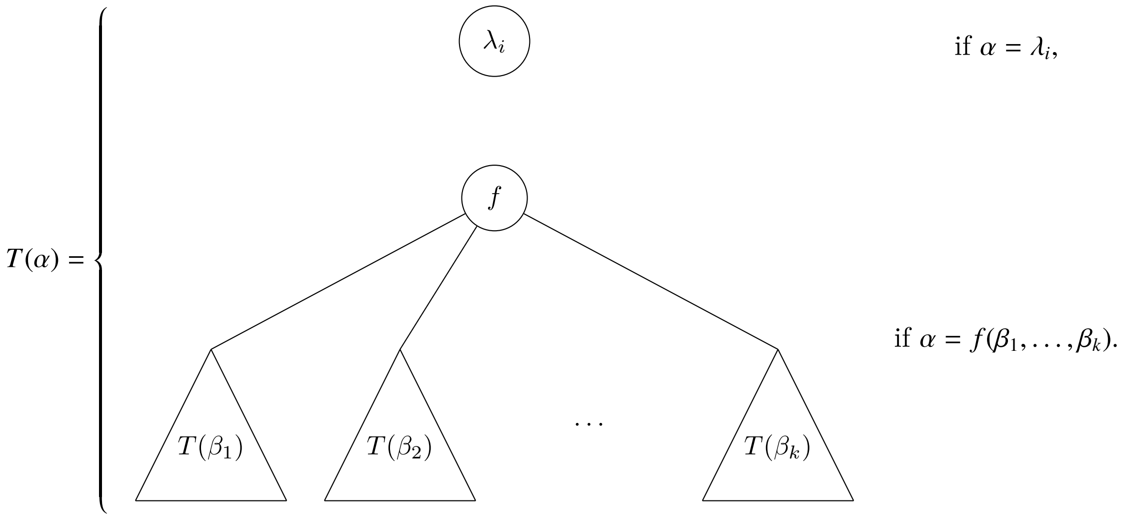
The answers to the above two questions can be quickly merged into the information we want. The only difference between these two questions is that for ∂D_i On the left $\lambda_i = 1$; For its right side there is $\lambda_i = 0$. For the other j $(j \neq i)$,∂D_i On the quilt D_j The covered part is $\lambda_j = 1$, Not D_j The covered part is $\lambda_j = 0$. The solutions to these two problems are also the same. Only the problems will be discussed below. 3a The solution.

Using the polar angle calculation defined in the previous chapter, ∂D_i The points on map to $[-\pi, \pi)$ Up, question 3a change into:

problem 3.1 $\forall i \in [1, n]$, Interval $[-\pi, \pi)$ Up to two given intervals make $\lambda_i = 1$, For other locations $\lambda_i = 0$. Ask for an envoy $F(\lambda_1, \ldots, \lambda_n) = 1$ For all positions, use $[-\pi, \pi)$ It is represented by a number of disjoint intervals.

Use the idea of scanning lines to make moving points P From $-\pi$ to π Move and maintain during the move $F(\lambda_1, \ldots, \lambda_n)$ Value. During the move there will be $O(n)$ An "event"-a certain λ_i The value of has changed,F The value is updated accordingly.F Value from 1 Changed to 0, And from 0 Changed to 1 The position of is the left and right end points of a certain interval.

Use the expression tree to display the structure of Boolean expressions. For a boolean expression α, Define its expression tree



In the first case,$T(\alpha)$ The only node is called the "input node"; the root node in the second case is called the "operation node",f The operator called the node. The expression tree is a rooted tree, and the input nodes are all leaf nodes. The children of the operation node are an ordered list, and the order of each parameter of the corresponding operator cannot be changed.

Build out α_F The expression tree, all the following discussion will be carried out on this tree. It is easy to verify that the length of an expression is equal to the number of nodes in its expression tree, so $T(\alpha_F)$ Have m Nodes. For each node u, Called subtree u

Means u And all its descendants form a subtree, subtree u A boolean expression is specified, sometimes the name of the node is also used u To refer to the expression, or the value of the expression.

when $\lambda_1, \dots, \lambda_n$ After the value is determined, define the node u Value of $W(u)$ Subtree u The value of the specified expression, namely

$$W(u) = \begin{cases} \lambda_i, & \text{if } u \text{ is an input node,} \\ f_u(W(v_1), W(v_2), \dots, W(v_k)), & \text{otherwise,} \end{cases}$$

among them $v_1, \dots, v_k$ Yes u The child node. In the second case,f_u Point u Operator. By repeatedly applying the above formula from bottom to top, the value of the root node can be calculated, namely $F(\lambda_1, \dots, \lambda_n)$.

when $\lambda_1, \dots, \lambda_n$ After the value is determined, the value of the input node is also determined. At this time, include the variable λ_i The input node of is equivalent to containing 0 The operation node of the meta-operator, the value of the node is either constant 0, Or Hengwei 1, Was λ_i The value of is determined. We represent them as operation nodes on the expression tree, so that the expression tree is the same as $\lambda_1, \dots, \lambda_n$ It's not directly related anymore. But for the sake of writing convenience, we still refer to these nodes as "contains λ_i Input node". When a certain λ_i When modified, include λ_i The operators of several input nodes need to be modified. Each node can be modified at most $O(1)$ Times. We need to solve the following problems:

problem 3.2 Design a data structure to support three operations:

(1) INITIALIZE-TREE. Initialize the data structure. Build an expression tree so that the value of all input nodes is 0.

(2) MODIFY-TREE(u, b). Modify input node u The value of, let $W(u) = b$.

(3) QUERY-TREE. Query the value of the root node $W(\text{root})$.

When a leaf is modified, the values of all its ancestors also need to be recalculated. In some special problems, the depth of the tree does not exceed $O(\log m)$, You can use the naive method to update the values of these nodes from the bottom to the top.

算法 1 数据结构的朴素实现

```
procedure INITIALIZE-TREE-SIMPLE
    for each node $u$ bottom-up do
        assume $u = f_u(v_1, \dots, v_k)$                ▷ 此处用结点名称指代其值
        let $u.processor$ be a new data structure of operator $f$
        $u.processor$.INITIALIZE
        for each $v_i$ do
            $u.processor$.MODIFY($i$, $W(v_i)$)
        end for
        $W(u) \leftarrow u.processor$.QUERY
    end for
end procedure

procedure MODIFY-TREE-SIMPLE($u$, $b$)
    $W(u) \leftarrow b$
    if $u =$ root then exit
    $p \leftarrow u.parent$
    assume $p = f_p(v_1, \dots, v_k)$ and $v_i = u$
    $p.processor$.MODIFY($i$, $b$)
    MODIFY-TREE-SIMPLE($p$, $p.processor$.QUERY)
end procedure

function QUERY-TREE-SIMPLE
    return $W$(root)
end function
```

The pseudo code above shows the naive implementation of the data structure.INITIALIZE-TREE-SIMPLE Create a data structure black box of its operator for each node *u.processor*, And set the parameters of the black box to the correct values. Subprocess MODIFY-TREE-SIMPLE Recursively visit the nodes that need to be recalculated in sequence. For each node, at most one of its parameters has changed. First 18,19 The black box is used to adjust the parameter to the correct value within the constant running time and obtain a new calculation result.INITIALIZE-TREE-SIMPLE The running time is $\Theta(m)$,MODIFY-TREE-SIMPLE The running time of is the upper bound of the maximum depth of the tree,QUERY-TREE-SIMPLE The running time is constant. If the maximum depth of the tree does not exceed $O(\log m)$, The algorithm is efficient enough.

In the case of a large tree depth, use tree chain division. For each node u, Remember s_u Subtree u The number of nodes. For non-leaf nodes u, If its children s The biggest node is v, Then called (u, v) Is the important side,v Yes u Heavy child$u$ The side to the other children is the light side, they are u The light child. The following properties are established:

- The heavy edges form several chains, and each node belongs to exactly one chain (the chain can have only one node). These chains are called heavy chains.
- The path from any node to the root passes at most $O(\log m)$ Light side, at most passing $O(\log m)$ A heavy chain.

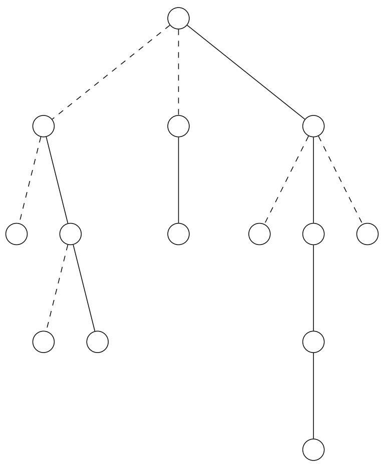


Figure 11

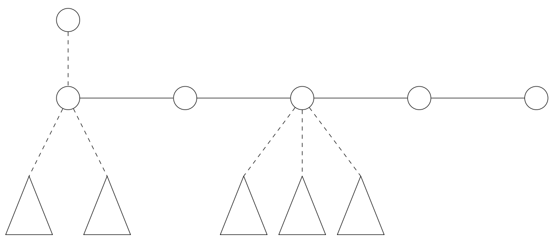


Figure 12

Figure 11 Shows the structure of a tree chain division. In the figure, the heavy side is represented by a solid line, and the light side is represented by a dashed line. Figure 12 It shows the structure of a heavy chain. In the figure, for clarity, light children are drawn below the chain, but in fact there is a fixed order between light children and heavy children.

For any node u, Assuming $u = f_u(v_1, \ldots, v_k)$,among them v_i It is its eldest son (ie, the next node on the heavy chain). Then, we can divide v_i Outside $k-1$ Parameters are regarded as constants, defining

$$g_u(v_i) = f_u(v_1, \ldots, v_i, \ldots, v_k),$$

Then $g_u(x)$ It is a univariate Boolean function. In particular, if f_u Yes 0 Meta, no parameters, then we add a dummy parameter xAnd make

$$g_u(x) = f_u(\ ),$$

at this time $g_u(x)$ Although it is a unary function in form, there are $g_u(0) = g_u(1)$ Yes, that is, the function value has nothing to do with the parameter value. In addition, define two unary functions f,g Compound

$$(f \circ g)(x) = f(g(x)),$$

It satisfies the associative law.

For a heavy chain, compound the chain in order g_u Will get a unary function g^*, It satisfies

$$g^*(0) = g^*(1) = W(r),$$

among them r It is the top of the chain, that is, the top node of the chain.g_u The definition directly illustrates the correctness of the above formula.

For each heavy chain, maintain a unary function with a line segment tree g_u Sequence and maintain the value at the top of the chain $W(r)$. When the leaf node 0 When the meta-operator is modified, its unary function on the heavy chain is also modified. Make this modification on this line segment tree, and re-query the composition of all unary functions to calculate $W(r)$ Value.$W(r)$ Updated,r The father of (if it exists) has a lighter child whose value has changed, causing its unary function to change, and the process is repeated further. algorithm 2 The pseudo code in shows an implementation.

In subprocess MODIFY-CHILD-VALUE In, we still use the black box of operators to calculate unary functions g_u, The running time of this sub-process is constant.*segment-tree*.QUERY The query is always the composite of all unary functions. The information has been stored in the root node of the line segment tree, so the running time is also constant. one more time MODIFY-TREE in,*segment-tree*.MODIFY Does not exceed $O(\log m)$, Each run time does not exceed $O(\log m)$, Therefore MODIFY-TREE Upper bound $O(\log^2 m)$.

On this basis, using the "globally balanced binary tree" instead of the line segment tree can make MODIFY-TREE The upper bound of running time drops to $O(\log m)$,see [1]. In addition,INITIALIZE-TREE The running time is $\Theta(m)$,QUERY-TREE The running time is $\Theta(1)$. We have got the problem 3.2 An efficient solution for.

Then, apply the above data structure to solve the problem 3.1. Before the "scan" starts, the data structure needs to be initialized $\Theta(m)$ time. In the process of "scanning", each leaf on the expression tree will be modified a constant number of times. The upper bound of the running time of this part is $O(m \log m)$. So the question 3.1 To $O(m \log m)$ The upper bound of the runtime is resolved.

At the beginning of this chapter, we put the question 3 change into $\Theta(n)$ Independent questions 3.1. So as to use the above method to get the problem 3 one of $O(nm \log m)$ The algorithm for the upper bound of the running time.

算法 2 数据结构的高效实现

```
procedure INITIALIZE-TREE
    INITIALIZE-TREE-SIMPLE
    build heavy-light decomposition
    calculate $g_u$   $(\forall u)$
    for each chain $c$ do
        let $c.segment\text{-}tree$ be a new segment tree of $g_u$
        initialize $c.segment\text{-}tree$
    end for
end procedure

procedure MODIFY-CHILD-VALUE($u$, $v$, $b$)
    assume $v_1, \ldots, v_k$ are children of $u$
    assume $v = v_i$, and $v_j$ is the heavy-child of $u$ $(i \neq j)$
    $u.processor$.MODIFY($i$, $b$)
    $u.processor$.MODIFY($j$, 0)
    $g_u(0) \leftarrow u.processor$.QUERY
    $u.processor$.MODIFY($j$, 1)
    $g_u(1) \leftarrow u.processor$.QUERY
end procedure

procedure RECALCULATE($u$)
    assume $u$ is the $i^{\text{th}}$ node of chain $c$
    $r \leftarrow c.top$
    $c.segment\text{-}tree$.MODIFY($i$, $g_u$)
    $g^* \leftarrow c.segment\text{-}tree$.QUERY
    $W(r) \leftarrow g^*(0)$                                ▷ 此时 $g^*(0) = g^*(1)$
    if $r \neq$ root then
        $p \leftarrow r.parent$
        MODIFY-CHILD-VALUE($p$, $r$, $W(r)$)
        RECALCULATE($p$)
    end if
end procedure

procedure MODIFY-TREE($u$, $b$)
    $g_u(0) \leftarrow b,\ g_u(1) \leftarrow b$
    RECALCULATE($u$)
end procedure

function QUERY-TREE
    return $W(\text{root})$
end function
```

5 Application and discussion

In this chapter, we discuss several variants, generalizations, and applications of the spherical compound area area problem.

Spherical polygon

A spherical polygon is an area surrounded by several spherical line segments connected end to end. In reality, it is often used to represent an area on the surface of the earth.

Our algorithm can handle spherical polygons. We can use the intersection of three "hemispheres" to represent a triangle, where the hemisphere refers to the area on one side of the great circle. Furthermore, the union of spherical triangles is used to represent spherical polygons. Here we do not repeatedly use the common binary "logical OR" operator, but directly use k Metalogical OR operator. In the above method, the spherical polygon is specified by a certain expression, and the depth of the expression tree is constant. After representing the spherical polygons, they can continue to participate in Boolean operations to form more complex complex regions.

Flat composite area

Different from practical applications, in algorithm competitions, the problem is often abstracted as an area problem on a plane. This is because the spherical geometry code is cumbersome to write and usually has little to do with the point of thinking, so it is omitted by abstracting it as a plane problem.

By combining the contents of Chapter 2 and Chapter 4, we can directly obtain an algorithm, which can calculate the area of the compound area specified by the Boolean operation of several circles on the plane. But this is not enough, because the "straight line" on the sphere is a special kind of circle, but not on the plane. To this end, in addition to the circle, we introduce another area-half plane.

Half plane Refers to the area to the left of a directed straight line. This directed line is the boundary of the half-plane and has the correct orientation.

Imitating the polar angle operation, we can define the**Dot product**Operation. Suppose the point set on a directed straight line is

$$\vec{a} + \vec{b}t \qquad (t \in \mathbb{R}),$$

among them t Increase in a straight line. Define a point on the line P Dot product

$$\mathrm{Dot}(P) = (P - \vec{a}) \cdot \vec{b},$$

On the right is the dot product of the two vectors. The dot product operation establishes all points on the directed line to the interval $(-\infty, +\infty)$ One-to-one mapping. Easy to verify, for any two areas D_i with D_j, They are all circles or half planes,D_j cover ∂D_i The part of is mapped to the number line with at most two intervals. In this way, we can continue to use the method of Chapter 4 to find the boundary of the composite area with the idea of scanning lines. ∂D^*. Correct ∂D^* The points are simple.

In this way, we have solved the "planar composite region area" problem with circles and half planes, and the running time has the same upper bound $O(nm \log m)$. It is easy to use a half-plane to represent a polygon on a plane, so I won't repeat it here. This method can be directly passed through three channels NOI Such problems that have occurred in:

- NOI 2004"Rainfall";
- NOI 2005"Lemon Tree Under the Moon";
- NOI 2009"Stroke".

Easy to calculate the boundary situation

In some questions, because of the special nature of the question, we can use the method for the particularity of the question to quickly calculate ∂D^*. Segment calculation ∂D^* The curve integral will not become the bottleneck of operating efficiency, because the running time of this part is proportional to ∂D^* The number of arcs in the middle, this is the previous stage (ie fast calculation ∂D^*)Output.

example 1"Lemon Tree Under the Moon"[2] On a given plane n Circle $A_1, \ldots, A_n$, Their centers are x On axis and centered x The coordinates are incremented. For two adjacent circles A_i versus A_{i+1}, If they do not contain each other, draw their two outer common tangent lines, and connect the four tangent points to form a trapezoid, and remember this trapezoidal area as B_i. definition

$$D^* = \bigcup_i A_i \cup \bigcup_i B_i,$$

Calculation D^* Area.

In this question,∂D^* on x Axisymmetric, just need to calculate x The part above the axis is denoted as C And ignore the direction. Put each circle A_i With the trapezoid on its left side B_{i-1}(If it exists) and record it as A'_i. Initial state order C Empty, join from left to right A'_i, And calculate the new boundary after each graphic is added C'. Easy to prove, each A'_i When joining, the original C Exactly one "suffix" (the continuous segment on the far right) is A'_i Overwritten and thus deleted and replaced by $\partial A'_i$ Some part of. Use the stack to maintain this process, you can $\Theta(n)$ Calculated within the running time CTo get ∂D^*.

In summary, we get one of the original problems $\Theta(n)$ Algorithm of running time. The algorithm has reached the lower bound of the time complexity theory of this question, which is much faster than the reference algorithm of the questioner $O(n^3)$.

DAG Formal boolean function

In Chapter 4, we use a tree structure to represent the λ_i To the calculation process of the final Boolean function. We use this structure to solve practical problems, largely relying on the flexibility of Boolean operators. For example, we know

$$a \oplus b = (a \vee b) \wedge \neg(a \wedge b),$$

That is, we can use basic logical operators $\neg,\wedge,\vee$ To define the "exclusive OR" operation. But the expanded form on the right does not work in our algorithm because a,b Each appears twice, and the tree structure cannot take the value of a node as the input of multiple operators. If the entire subtree is copied, the time complexity will rise to an exponential level, which is unacceptable. The method we take is to directly define an "exclusive OR" operator, and handle the reuse of intermediate variables within this operator.

The above method can only handle "local" variable reuse. In computer-aided design, the following situation often occurs: the entire calculation process is given by the user, and the results of each calculation are stored as intermediate variables, which can be accessed at will in the future. For such a highly free calculation process, we can only use DAG(Directed acyclic graph) to represent it. In the figure, each node represents an intermediate variable, which is the direct result of an operator, and each parameter of the operator corresponds to each entry edge of the node. Here, the incoming edge of a node is an ordered list, and the order of each parameter of the corresponding operator cannot be changed at will.

By sacrificing some operational efficiency, our algorithm can handle this situation. assumed DAG The number of sides is m. Then when $\lambda_1, \ldots, \lambda_n$ After the value of is determined, the final Boolean function F The value can be in $O(m)$ Calculated within the time. We no longer use data structures to maintain F Value, but after a variable changes, with $O(m)$ Time is completely recalculated F value. The time complexity of the entire algorithm is $O(n^2m)$.

example 2"Calculating Volume"[3] enter n Instructions, each instruction is one of the following:

(1) Create a ball

(2) Create a cuboid with each side parallel to the coordinate axis;

(3) Enter the two instruction numbers before this instruction i, j, Create a three-dimensional figure for the first i Directives and Articles j Handover of instructions/and/difference.

For each command, output the volume of the three-dimensional figure it creates.

Process all graphics together after offline. The data scale, value range, and accuracy requirements of this question are relatively low. You can use numerical integration for one dimension to transform the problem into a two-dimensional problem. The input to this two-dimensional problem contains only circles and rectangles, and the calculation is determined by DAG Gives,DAG Share $O(n)$ Edge. In particular, we need to ask for DAG Every node u Compound area specified by the value of D_u^* The area of ??, that is, all nodes are output nodes. When a variable λ_i When changed, recalculate DAG The value of all the nodes on the total need $O(n)$ Time, so solving the two-dimensional sub-problem requires $O(n^3)$ time. The time complexity of solving the original problem is $O(n^3 I)$,among them I Is the number of times the numerical integration method calls the evaluation function.

Surface area of the sphere

example 3 In a given space n Spheres, calculate the surface area of their union.

For a sphere, the part that is not covered by other balls becomes the surface of the sphere. For each sphere, calculate the area of the uncovered part, that is, the sphere

$$D^* = U - \bigcup_i D_i$$

Area, where U Is the entire sphere,D_i Is the sphere i The area covered by a ball is a ball cap. This turns the problem into a problem of the area of the spherical compound area, and obtains a $O(n^3 \log n)$ Algorithm.

Another interesting problem is to calculate the volume of the ball union. Although the above method can find the boundary of the ball union, the author has not found an accurate algorithm for calculating the volume of the ball union. This issue remains to be studied further.

6 to sum up

This article introduces the problem of spherical compound area area, and proposes Green's formula method to solve it. This problem has a wide range of applications in reality, mainly in the area of the earth's surface. Spherical geometry is not suitable for the propositions of algorithm competitions due to its lengthy code and numerous details, but our algorithm can be modified to adapt to the plane situation, thus reflecting its advantages in algorithm competitions.

However, the research on this issue is not over yet. In practical applications, if we need a more accurate area of a region, we should abstract the surface of the earth as an ellipsoid, so we need to solve the problem of the area of the composite ellipsoid. In addition, in Chapter 5 DAG Boolean function of the form, we only get a running time upper bound as $O(n^2m)$ The algorithm, its operating efficiency may have room for improvement. The above two related issues need to be further studied.

Thanks

Thanks to Zhou Hangrui for providing "Lemon Tree under the Moon"[2]Realization of the linear algorithm and its program, and assisted the drawing of some pictures.

Thank you Xu Zhean for discussing the application of Green's formula method with me.

references

[1] Yang Zhe. SPOJ375 QTREE Some research on the solution. Homework of the National Training Team of the Chinese Informatics Olympiad, 2007: 9-11.

[2] Hu Weidong. Lemon tree under the moon. National Youth Informatics Olympiad, 2005.
https://www.luogu.com.cn/problem/P4207.

[3] Xu Mingkuan. Two-dimensional computational geometry related algorithms and practical applications. National Youth Informatics Olympic Winter Camp, 2018: 70-72.

[4] people's education Press, Course Materials Research Institute, Experimental Research Group of Middle School Mathematics Textbook. General High School Curriculum Standard Experimental Textbook Mathematics Elective 3-3(B Version): Geometry on the sphere[M]. people's education Press, 2008: 27-28.

[5] Chang Gengzhe, Shi Jihuai. Mathematical Analysis Course[M]. First 3 Version. University of Science and Technology of China Press, 2012.

[6] Area. ACM/ICPC Asia Regional Nanjing Online, 2013.
http://acm.hdu.edu.cn/showproblem.php?pid=4748.

[7] Todhunter I. *Spherical Trigonometry, for the use of colleges and schools: with numerous examples*[M]. Macmillan, 1863: 67-71.