Standard Code Library

Tempest

October, 2014

Contents

Chapter 1

数学

1.1 平面几何公式

三角形

1. 半周长 $P = (a + b + c)/2$
2. 面积 $S = aH_a/2 = ab\sin(C)/2 = \sqrt{P(P-a)(P-b)(P-c)}$
3. 中线 $M_a = \sqrt{2(b^2+c^2)-a^2}/2 = \sqrt{b^2+c^2+2bc\cos(A)}/2$
4. 角平分线 $T_a = \sqrt{bc((b+c)^2-a^2)}/(b+c) = 2bc\cos(A/2)/(b+c)$
5. 高线 $H_a = b\sin(C) = c\sin(B) = \sqrt{b^2-((a^2+b^2-c^2)/(2a))^2}$
6. 内切圆半径

$$
\begin{aligned}
r &= S/P = \mathrm{arcsin}(B/2)\sin(C/2)/\sin((B+C)/2) = 4R\sin(A/2)\sin(B/2)\sin(C/2) \\
&= \sqrt{(P-a)(P-b)(P-c)/P} = P\tan(A/2)\tan(B/2)\tan(C/2)
\end{aligned}
$$

7. 外接圆半径 $R = abc/(4S) = a/(2\sin(A)) = b/(2\sin(B)) = c/(2\sin(C))$

四边形

$D1, D2$ 为对角线，M 对角线中点连线，A 为对角线夹角

1. $a^2 + b^2 + c^2 + d^2 = D1^2 + D2^2 + 4M^2$
2. $S = D1D2\sin(A)/2$
3. 圆内接四边形 $ac + bd = D1D2$
4. 圆内接四边形，P 为半周长 $S = \sqrt{(P-a)(P-b)(P-c)(P-d)}$

正 n 边形

R 为外接圆半径,r 为内切圆半径

1. 中心角 $A = 2\pi/n$
2. 内角 $C = (n-2)\pi/n$
3. 边长 $a = 2\sqrt{R^2 - r^2} = 2R\sin(A/2) = 2r\tan(A/2)$
4. 面积 $S = nar/2 = nr^2\tan(A/2) = nR^2\sin(A)/2 = na^2/(4\tan(A/2))$

圆

1. 弧长 $l = rA$
2. 弦长 $a = 2\sqrt{2hr - h^2} = 2r\sin(A/2)$
3. 弓形高 $h = r - \sqrt{r^2 - a^2/4} = r(1-\cos(A/2)) = \arctan(A/4)/2$
4. 扇形面积 $S1 = rl/2 = r^2A/2$
5. 弓形面积 $S2 = (rl - a(r-h))/2 = r^2(A - \sin(A))/2$

棱柱

1. 体积 $V = Ah$，A 为底面积，h 为高
2. 侧面积 $S = lp$，l 为棱长，p 为直截面周长
3. 全面积 $T = S + 2A$

棱锥

1. 体积 $V = Ah$，A 为底面积，h 为高
2. 正棱锥侧面积 $S = lp$，l 为棱长，p 为直截面周长
3. 正棱锥全面积 $T = S + 2A$

棱台

1. 体积 $V = (A1 + A2 + \sqrt{A1A2})h/3$, $A1, A2$ 为上下底面积，h 为高
2. 正棱台侧面积 $S = (p1 + p2)l/2$，$p1, p2$ 为上下底面周长，l 为斜高
3. 正棱台全面积 $T = S + A1 + A2$

圆柱

1. 侧面积 $S = 2\pi rh$
2. 全面积 $T = 2\pi r(h + r)$
3. 体积 $V = \pi r^2 h$

圆锥

1. 母线 $l = \sqrt{h^2 + r^2}$
2. 侧面积 $S = \pi r l$
3. 全面积 $T = \pi r(l + r)$
4. 体积 $V = \pi r^2 h/3$

圆台

1. 母线 $l = \sqrt{h^2 + (r1 - r2)^2}$
2. 侧面积 $S = \pi(r1 + r2)l$
3. 全面积 $T = \pi r1(l + r1) + \pi r2(l + r2)$
4. 体积 $V = \pi(r1^2 + r2^2 + r1r2)h/3$

球

1. 全面积 $T = 4\pi r^2$
2. 体积 $V = 4\pi r^3/3$

球台

1. 侧面积 $S = 2\pi r h$
2. 全面积 $T = \pi(2rh + r1^2 + r2^2)$
3. 体积 $V = \pi h(3(r1^2 + r2^2) + h^2)/6$

球扇形

1. 全面积 $T = \pi r(2h + r0)$，h 为球冠高，$r0$ 为球冠底面半径
2. 体积 $V = 2\pi r^2 h/3$

1.2 NTT

```
const int modulo(786433);
const int G(10);//原根
int pw[999999];
void FFT(int P[], int n, int oper) {
    for(int i(1), j(0); i < n - 1; i++) {
        for(int s(n); j ^= s >>= 1, ~j & s;);
        if (i < j)
            swap(P[i], P[j]);
    }
    int unit_p0;
    for(int d(0); (1 << d) < n; d++) {
```

```
        int m(1 << d), m2(m * 2);
        unit_p0 = oper == 1?pw[(modulo - 1) / m2]:pw[modulo - 1 - (modulo - 1) / m2];
        for(int i = 0; i < n; i += m2) {
            int unit(1);
            for(int j(0); j < m; j++) {
                int &P1 = P[i + j + m], &P2 = P[i + j];
                int t = (long long)unit * P1 % modulo;
                P1 = (P2 - t + modulo) % modulo;
                P2 = (P2 + t) % modulo;
                unit = (long long)unit * unit_p0 % modulo;
            }
        }
    }
}

int nn;
int A[N], B[N], C[N];
//A * B = C;
//len = nn
void multiply() {
    FFT(A, nn, 1);
    FFT(B, nn, 1);
    for(int i(0); i < nn; i++) {
        C[i] = (long long)A[i] * B[i] % modulo;
    }
    FFT(C,  nn, -1);
}

int main() {
    pw[0] = 1;
    for(int i(1); i < modulo; i++) {
        pw[i] = (long long)pw[i - 1] * G % modulo;
    }
}
```

1.3 FFT

```
void FFT(Complex P[], int n, int oper) {
    for (int i(1), j(0); i < n - 1; i++) {
        for (int s(n); j ^= s >>= 1, ~j & s;);
        if (i < j)
            swap(P[i], P[j]);
    }
    Complex unit_p0;
    for (int d(0); (1 << d) < n; d++) {
        int m(1 << d), m2(m * 2);
        double p0(pi / m * oper);
        unit_p0.imag(sin(p0));
```

```
        unit_p0.real(cos(p0));
        for (int i(0); i < n; i += m2) {
            Complex unit = 1;
            for (int j = 0; j < m; j++) {
                Complex &P1 = P[i + j + m], &P2 = P[i + j];
                Complex t = unit * P1;
                P1 = P2 - t;
                P2 = P2 + t;
                unit = unit * unit_p0;
            }
        }
    }
}
void multiply() {
    FFT(a, n, 1);
    FFT(b, n, 1);
    for(int i(0); i < n; i++) {
        c[i] = a[i] * b[i];
    }
    FFT(c, n, -1);
    for(int i(0); i < n; i++) {
        ans[i] += (int)(c[i].real() / n + 0.5);
    }
}
```

1.4 中国剩余定理

包括扩展欧几里得，求逆元，和保证除数互质条件下的 CRT

```
LL x, y;
void exGcd(LL a, LL b)
{
    if (b == 0) {
        x = 1;
        y = 0;
        return;
    }
    exGcd(b, a % b);
    LL k = y;
    y = x - a / b * y;
    x = k;
}

LL inversion(LL a, LL b)
{
    exGcd(a, b);
    return (x % b + b) % b;
}

```