

ICPC 対策資料集  
(引用文献示してないですごめんなさい)

杉本 元気

平成 24 年 7 月 1 日



# 第1章 計算量

## 1.1 実行時間の目安

実行制限時間が1秒の場合

1,000,000( $10^6$ )	余裕
10,000,000( $10^7$ )	おそらく間に合う
100,000,000( $10^8$ )	非常にシンプルな処理でない限り厳しい

## 1.2 オーダー

Nの目安

オーダー	$10^6$	$10^7$	$10^8$
$O(N \log N)$	62,746	526,172	4,523,070
$O(N^2)$	1000	3162	10000
$O(N^3)$	100	215	464
$O(2^N)$	20	23	26
$O(N!)$	9	10	11



## 第2章 最初に作るもの

### 2.1 Makefile

```
1 all: a b c d e
2
3 a: a.cpp
4   g++ -O2 a.cpp -o a
5 b: b.cpp
6   g++ -O2 b.cpp -o b
7 c: c.cpp
8   g++ -O2 c.cpp -o c
9 d: d.cpp
10  g++ -O2 d.cpp -o d
11 e: e.cpp
12  g++ -O2 e.cpp -o e
```

### 2.2 template.cpp

```
1 #include <stdio.h>
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     return 0;
8 }
```

### 2.3 copy.sh

template.cpp を一気にコピー

```
1 for f in a b c d e
2 do
3   cp template.cpp $f.cpp
4 done
```

### 2.4 vimrc

```
1 set foldmethod=marker
2 set tabstop=4
3 set shiftwidth=4
4
5 set smartindent
6 set smarttab
7 set smartcase
8
9 syntax enable
10 colorscheme desert
11 set background=dark
12
13 set cursorline
```

## 第3章 ヘルパー

### 3.1 定数

`M_PI` ( $\pi$ ), `M_PI_2` ( $\pi/2$ ), `M_PI_4` ( $\pi/4$ ) が `cmath` で定義されてるからそれでも OK (%.15lf まで正確)。

```
1 static const double PI (3.14159265358979323846);
2 static const double EPS (1e-10);
```

### 3.2 言語機能

#### 3.2.1 変数型

```
1 typedef long long int ll;
2 typedef unsigned long long int ull;
3 typedef vector<int> vi;
4 typedef vector<vi> vvi;
5 typedef vector<string> vs;
6 typedef pair<int, int> pii;
```

#### 3.2.2 conversion

```
1 inline int toInt(string s) {int v; stringstream sin(s); sin>>v;
   return v;}
2 template<class T> inline string toString(T x) {ostringstream
   sout; sout<<x; return sout.str();}
```

#### 3.2.3 ループ

```
1 #define FOR(i, b, e) for (typeof(b) i = (b); assert((i) <= (e)),
   i != (e); ++i)
2 #define REP(i, n) FOR(i, 0, n)
```

### 3.2.4 オペレーターオーバーロード

```
1 #define OPOVER(_op, _type) inline bool operator _op (const _type
    &t) const
```

### 3.2.5 配列の要素数

```
1 #define arrsz(a) ( sizeof(a) / sizeof(a[0]) )
```

## 3.3 pair

```
1 #define F first
2 #define S second
3 #define MP(a, b) make_pair(a, b)
```

## 3.4 vector

```
1 #define SZ(a) ((ll)a.size())
2 #define PB(e) push_back(e)
3 #define SORT(v) sort((v).begin(), (v).end())
4 #define RSORT(v) sort((v).rbegin(), (v).rend())
```

## 3.5 イテレーション

```
1 #define ALL(a) (a).begin(), (a).end()
2 #define RALL(a) (a).rbegin(), (a).rend()
3 #define EACH(t,i,c) for(t::iterator i=(c).begin(); i!=(c).end();
    ++i)
4 #define EXIST(s,e) ((s).find(e)!=(s).end())
```

## 3.6 bit 操作

```
1 #define BIT(n) (1ULL << (n))
2 #define BITOF(n, m) ((n) >> (m) & 1)
```

## 3.7 bool 判定

### 3.7.1 $a \leq b \leq c$

```
1 #define RANGE(a, b, c) ((a) <= (b) && (b) <= (c))
```



## 第4章 幾何

### 4.1 幾何ライブラリ

```

1 typedef complex<double> P;
2
3 // 許容する誤差
4 #define EPS (1e-10)
5 // 2つのスカラーが等しいかどうか
6 #define EQ(a,b) (abs((a)-(b)) < EPS)
7 // 2つのベクトルが等しいかどうか
8 #define EQV(a,b) ( EQ((a).real(), (b).real()) && EQ((a).imag(),
9                  (b).imag()) )
10
11 // ベクトル a の絶対値を求める
12 double length = abs(a);
13
14 // 2点 a, b 間の距離を求める
15 double distance = abs(a-b);
16
17 // ベクトル a の単位ベクトルを求める
18 P b = a / abs(a);
19
20 // ベクトル a の法線ベクトル n1, n2 を求める
21 P n1 = a * P(0, 1);
22 P n2 = a * P(0, -1);
23
24 // ベクトル a の単位法線ベクトル un1, un2 を求める
25 P un1 = (a * P(0, +1)) / abs(a);
26 P un2 = (a * P(0, -1)) / abs(a);
27
28 // 内積 (dot product) :  $a \cdot b = |a||b|\cos$ 
29 double dot(P a, P b) {
30     return (a.real() * b.real() + a.imag() * b.imag());
31 }
32 double getcos(P a, P b) {
33     return ( dot(a, b) / ( abs(a) * abs(b) ) );
34 }
35
36 // 外積 (cross product) :  $a \times b = |a||b|\sin$ 
37 double cross(P a, P b) {
38     return (a.real() * b.imag() - a.imag() * b.real());
39 }
40 double getsin(P a, P b) {
41     return ( cross(a, b) / ( abs(a) * abs(b) ) );
42 }
43
44 // 2直線の直交判定 :  $a \perp b \iff \text{dot}(a, b) = 0$ 
45 int is_orthogonal(P a1, P a2, P b1, P b2) {
46     return EQ( dot(a1-a2, b1-b2), 0.0 );
47 }

```

```

48 // 2直線の平行判定 :  $a//b \Leftrightarrow \text{cross}(a, b) = 0$ 
49 int is_parallel(P a1, P a2, P b1, P b2) {
50     return EQ( cross(a1-a2, b1-b2), 0.0 );
51 }
52
53 // 点cが直線a, b上にあるかないか
54 int is_point_on_line(P a, P b, P c) {
55     return EQ( cross(b-a, c-a), 0.0 );
56 }
57
58 // 点cが線分a, b上にあるかないか (1)
59 int is_point_on_ls(P a, P b, P c) {
60     return EQ( cross(b-a, c-a), 0.0 ) &&
61         (dot(b-a, c-a) > -EPS) &&
62         (dot(a-b, c-b) > -EPS);
63 }
64
65 // 点cが線分a, b上にあるかないか (2)
66 int is_point_on_ls(P a, P b, P c) {
67     //  $|a-c| + |c-b| \leq |a-b|$  なら線分上
68     return (abs(a-c) + abs(c-b) < abs(a-b) + EPS);
69 }
70
71 // 点a, bを通る直線と点cとの距離
72 double distance_l_p(P a, P b, P c) {
73     return abs(cross(b-a, c-a)) / abs(b-a);
74 }
75
76 // 点a, bを端点とする線分と点cとの距離
77 double distance_ls_p(P a, P b, P c) {
78     if ( dot(b-a, c-a) < EPS ) return abs(c-a);
79     if ( dot(a-b, c-b) < EPS ) return abs(c-b);
80     return abs(cross(b-a, c-a)) / abs(b-a);
81 }
82
83 // a1, a2を端点とする線分とb1, b2を端点とする線分の交差判定
84 int is_intersected_ls(P a1, P a2, P b1, P b2) {
85     return ( cross(a2-a1, b1-a1) * cross(a2-a1, b2-a1) < EPS ) &&
86         ( cross(b2-b1, a1-b1) * cross(b2-b1, a2-b1) < EPS );
87 }
88
89 // a1, a2を端点とする線分とb1, b2を端点とする線分の交点計算
90 P intersection_ls(P a1, P a2, P b1, P b2) {
91     P b = b2-b1;
92     double d1 = abs(cross(b, a1-b1));
93     double d2 = abs(cross(b, a2-b1));
94     double t = d1 / (d1 + d2);
95
96     return a1 + (a2-a1) * t;
97 }
98
99 // a1, a2を通る直線とb1, b2を通る直線の交差判定
100 int is_intersected_l(P a1, P a2, P b1, P b2) {
101     return !EQ( cross(a1-a2, b1-b2), 0.0 );
102 }
103
104 // a1, a2を通る直線とb1, b2を通る直線の交点計算
105 P intersection_l(P a1, P a2, P b1, P b2) {
106     P a = a2 - a1; P b = b2 - b1;
107     return a1 + a * cross(b, b1-a1) / cross(b, a);
108 }
109

```

```

110 // 直線 ab と円  $|x-c|=r$  の交点を求める
111 void intersection_l_c(P a, P b, P c, double r, P ans[]){
112     if(distance_l_p(a, b, c) > r + EPS) return;
113     P v = (b-a) / abs(b-a);
114     double delta = dot(v, a-c) * dot(v, a-c) - abs(a-c) * abs(a-c)
115         + r * r;
116     double t = -dot(v, a-c);
117     double s = sqrt(delta);
118     ans[0] = a + v * (t + s);
119     ans[1] = a + v * (t - s);
120 }
121 // x (y=k) に関する対象変換 k=0で x 軸による変換
122 P x_translate(P t, double k){
123     return P(t.x, 2*k - t.y);
124 }
125
126 // y (x=k) に関する対象変換 k=0で y 軸による変換
127 P y_translate(P t, double k){
128     return P(2*k - t.x, t.y);
129 }
130
131 // 点 P k に関する対象変換 P(0,0)で原点による変換
132 P o_translate(P t, P k){
133     return k + (k - t);
134 }
135
136 // 点 p を中心として r (radian) 回転 p(0,0)で原点を中心として回転
137 P rotate(P t, P p, double r){
138     // double r=radians(angle);
139     return (t - p) * P(cos(r), sin(r)) + p;
140 }
141
142 // 2円  $|x-a|=ra$  と  $|x-b|=rb$  の交点計算
143 bool intersection_c_c(P a, double ra, P b, double rb, P ans[]){
144     double di = abs(a-b);
145     if(di > ra+rb || di < abs(ra-rb)) return false;
146     double t=(ra*ra - rb*rb + di*di) / (di+di);
147     double rd = acos(t/ra);
148
149     P dv = (b-a) / abs(b-a);
150     P g1 = rotate(dv, P(0,0), rd);
151     P g2 = rotate(dv, P(0,0), -rd);
152     ans[0] = a + g1*ra;
153     ans[1] = a + g2*ra;
154     return true;
155 }
156
157 // 点 p から直線 ab までの法線ベクトルを求める
158 P normal_vector(P a, P b, P p){
159     P v = (b-a) / abs(b-a);
160     v = cross(v, p-a) > 0 ? v * P(0, -1) : v * P(0, 1);
161     return v * distance_l_p(a, b, p);
162 }
163
164 // 直線 ab に関する対象変換
165 P f_translate(P t, P a, P b){
166     return t + normal_vector(a, b, t) * 2;
167 }
168
169 // 3角形の面積を求める
170 double area_triangle(P a, P b, P c){

```

```
171 |     return abs(cross(c-a, b-a) * 0.5);
172 | }
173 |
174 | // 多角形の面積を求める // 凸包のソート済みが前提
175 | double area_polygon(vector<P> t){
176 |     double ans = 0.0;
177 |     for(unsigned int i = 0; i < t.size(); i++)
178 |         ans += cross(t[i], t[(i+1) % t.size()]);
179 |     return ans / 2;
180 | }
```

## 第5章 ハマったときは

### 5.1 Wrong Answer

#### 5.1.1 チェックリスト

- 条件分岐  
if, while, for, switch で検索して条件を吟味

#### 5.1.2 よくあるミス

- 初期化し忘れてない?  
vector は clear しないと。

### 5.2 Runtime Error

#### 5.2.1 不正なメモリアクセス

- 配列宣言時の要素数は正しい?  
“int field[10][20]” であるべきところが “int field[20][10]” になってたり。

### 5.3 デバッグがうまくいかない

- 出力フォーマットミスってない?  
double を %d で表示しようとしてたり