

## Fudan University ICPC Team 7 Notebook (2016-17)

hzf,phz,ck

## Contents

September 25, 2016

1	Math
1.1	Simpson Integration . . . . .
2	Data Structure
2.1	Divide And Conquer on Tree . . . . .
3	Algorithm
4	other

## 1 Math

## 1.1 Simpson Integration

Let  $f()$  be the integral function, calculate integration in interval  $[a, b]$  by calling `asr(a,b,eps)`.

```

1 typedef double DB;
2 DB f(DB x) { // integral function
3     return log(x*x+2.24234*x-M_PI);
4 }
5 DB simpson(DB a, DB b) {
6     return (b-a)*(f(a)+4.0*f((a+b)/2.0)+f(b))/6.0;
7 }
8 DB asr(DB l, DB r, DB eps, DB res) {
9     DB m = (l+r)/2.0;
10    DB ls = simpson(l, m), rs = simpson(m, r);
11    if (fabs(ls+rs-res) < eps*15) return ls+rs+(ls+rs-res)/15;
12    return asr(l, m, eps/2.0, ls)+asr(m, r, eps/2.0, rs);
13 }
14 DB asr(DB l, DB r, DB eps) {
15     return asr(l, r, eps, simpson(l, r));
16 }
17 int main() {
18     printf("%.12lf\n", asr(1, 10, 1e-2));
19     return 0;
20 }

```

## 2 Data Structure

## 2.1 Divide And Conquer on Tree

- `t_dac::init();`

- `t_dac::addEdge();`
- `t_dac::solve();`

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int INF = 0x3f3f3f3f;
5 typedef long long LL;
6
7 namespace t_dac {
8
9     const int maxV = 100000 + 100;
10    const int maxE = maxV * 2;
11
12    int d, t;
13    bool vis[maxV];
14
15    LL ans;
16
17    int h[maxV], to[maxE], nxt[maxE], wei[maxE], e;
18    void addEdge(int u, int v, int c) {
19        nxt[e] = h[u], to[e] = v, wei[e] = c, h[u] = e++;
20        nxt[e] = h[v], to[e] = u, wei[e] = c, h[v] = e++;
21    }
22
23    void init() {
24        memset(h, -1, sizeof h);
25        e = 0;
26        memset(vis, false, sizeof vis);
27        ans = 0;
28    }
29    int get_s(int u, int f) {
30        int r = 1;
31        for (int i = h[u]; i != -1; i = nxt[i]) {
32            int v = to[i];
33            if (!vis[v] && v != f)
34                r += get_s(v, u);
35        }
36        return r;
37    }
38    int get_d(int u, int f, int s) {
39        int r = 1, mx = 0, w;
40        for (int i = h[u]; i != -1; i = nxt[i]) {
41            int v = to[i];
42            if (!vis[v] && v != f) {
43                r += (w = get_d(v, u, s));
44                mx = max(mx, w);
45            }
46        }
47        mx = max(mx, s - r);
48        if (mx < t) {
49            t = mx;
50            d = u;
51        }
52        return r;
53    }
54
55    void calc(int u);

```

```

56
57 void dfs(int u) {
58
59     int s = get_s(u, -1);
60     t = INF, d = u;
61     get_d(u, -1, s);
62     //fprintf(stderr, "%d %d\n", u, d);
63     calc(d);
64
65     vis[d] = 1;
66     for (int i = h[d]; i != -1; i = nxt[i]) {
67         int v = to[i];
68         if (!vis[v])
69             dfs(v);
70     }
71 }
72
73 void solve() {
74     dfs(1);
75 }
76
77
78 };

```

### 3 Algorithm

#### 4 other

sdfasdfasd

orz wenxy

$$\sum_{i=1}^{10} i$$

```

1 #include <cstdio>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 int main() {
6     return 0;
7 }

```

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     return 0;
6 }

```