1.10

洛谷题单

P1923

#include <iostream>

using namespace std;

int Partition(int a[], int low, int high)

{

int pivotkey;

a[0] = a[low];

pivotkey = a[low];

while (low < high) {

while (low < high && a[high] >= pivotkey)--high;

a[low] = a[high];

while (low < high && a[low] <= pivotkey)++low;

a[high] = a[low];

}

a[low] = a[0];

return low;

}

void quicksort(int a[], int low, int high) {

if (low < high) {

int pivotloc = Partition(a, low, high);

quicksort(a, low, pivotloc - 1);

quicksort(a, pivotloc + 1, high);

}

}

int main()

{

int n, a[1000], a1 = 1;

cin >> n;

for (int i = 1; i <= n; i++) {

cin >> a[i];

}

quicksort(a, a1, n - 1);

for (int i = 1; i <= n; i++)

cout << a[i]<<" ";

return 0;

}

P2240部分背包问题

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

double t, v[101], w[101], a[101], ans;

cin >> n >> t;

for (int i = 1; i <= n; i++)

{

cin >> w[i] >> v[i];

a[i] = v[i] / w[i];

}

for (int i = 1; i <= n; i++)

{

for (int j = 1; j < n; j++)

{

if (a[j] < a[j + 1])

{

swap(a[j], a[j + 1]);

swap(v[j], v[j + 1]);

swap(w[j], w[j + 1]);

}

}

}

for (int i = 1; i <= n; i++)

{

if (t - w[i] > -0.000001)

{

t -= w[i];

ans += v[i];

}

else

{

ans += t \* a[i];

break;

}

}

printf("%.2lf", ans);

return 0;

P1106

删数问题

#include<bits/stdc++.h>

using namespace std;

long long p, pp, xiao, q, ppp, mins;

string n, st2;

int main()

{

cin >> n;

cin >> p;

pp = n.size();

for (int i = 1; i <= pp - p; i++)

{

ppp = pp - p - i;

xiao = 2100000000;

for (int j = q; j < pp - ppp; j++)

{

if (n[j] - 48 < xiao)

{

xiao = n[j] - 48;

q = j + 1;

}

}

mins = mins \* 10 + xiao;

}

cout << mins;

return 0;

}

P2249

查找

#include<iostream>

#include<map>

using namespace std;

map<int, int>a;

int main()

{

int m, n, k, l = -1;

cin >> n >> m;

for (int i = 1; i <= n; i++)

{

cin >> k;

if (l != k) {

a[k] = i;

l = k;

}

}

for (int i = 0; i < m; i++)

{

cin >> k;

if (a.count(k))cout << a[k] << ' ';

else cout << -1 << ' ';

}

return 0;

}

P1102

数对

#include<bits/stdc++.h>

using namespace std;

int n, c;

long long ans;

map<int, int> a;

int num[200005];

int main()

{

scanf("%d%d", &n, &c);

for (int i = 1; i <= n; i++)

{

scanf("%d", &num[i]);

a[num[i]]++;

}

for (int i = 1; i <= n; i++)

{

ans += a[num[i] + c];

}

printf("%lld", ans);

return 0;

}

P1024

方程求解

#include<iostream>

#include<cstdio>

using namespace std;

double a, b, c, d, a1, b1, c1, d1;

int num;

int main()

{

scanf("%lf%lf%lf%lf", &a, &b, &c, &d);

for (double i = -100.00; i <= 100.00; i += 0.001)

{

double l = i, r = i + 0.001;

if ((a \* l \* l \* l + b \* l \* l + c \* l + d) \* (a \* r \* r \* r + b \* r \* r + c \* r + d) < 0)

printf("%.2f ", l), num++;

if (num == 3) break;

}

return 0;

}

P3366

最小生成树

#include<bits/stdc++.h>

const int maxn = 1e6 + 1;

inline int read()

{

register int x = 0, ch = getchar(), f = 1;

while (!isdigit(ch)) { if (ch == '-') f = -1; ch = getchar(); }

while (isdigit(ch)) x = x \* 10 + ch - '0', ch = getchar();

return x \* f;

}

int n, m;

struct node {

int u;

int v;

int w;

}e[maxn];

int fa[maxn], cnt, sum, num;

void add(int x, int y, int w)

{

e[++cnt].u = x;

e[cnt].v = y;

e[cnt].w = w;

}

bool cmp(node x, node y)

{

return x.w < y.w;

}

int find(int x)

{

return fa[x] == x ? fa[x] : fa[x] = find(fa[x]);

}

void kruskal()

{

for (int i = 1; i <= cnt; i++)

{

int x = find(e[i].u);

int y = find(e[i].v);

if (x == y) continue;

fa[x] = y;

sum += e[i].w;

if (++num == n - 1) break;

}

}

int main()

{

n = read();

m = read();

for (int i = 1; i <= n; i++) fa[i] = i;

while (m--)

{

int x, y, w;

x = read();

y = read();

w = read();

add(x, y, w);

}

std::sort(e + 1, e + 1 + cnt, cmp);

kruskal();

printf("%d", sum);

return 0;

}

P1195

口袋的天空

#include<bits/stdc++.h>

using namespace std;

int n, k, m;

int fa[1000050];

struct node {

int x;

int y;

int l;

} a[1000005];

int cmp(const void\* a, const void\* b) {

struct node\* c = (node\*)a;

struct node\* d = (node\*)b;

return c->l - d->l;

}

int find(int x)

{

if (x != fa[x])

fa[x] = find(fa[x]);

return fa[x];

}

void work(int x, int y)

{

x = find(x);

y = find(y);

if (x == y)

return;

fa[x] = y;

}

int main() {

cin >> n >> m >> k;

for (int i = 0; i < m; i++) {

scanf("%d%d%d", &a[i].x, &a[i].y, &a[i].l);

}

qsort(a, m, sizeof(a[0]), cmp);

for (int i = 1; i <= n; i++)

fa[i] = i;

int num = n - k;

int ans = 0;

for (int i = 0; i < m; i++)

{

if (num == 0)

break;

int aaa = find(a[i].x);

int wzx = find(a[i].y);

if (aaa != wzx)

{

work(a[i].x, a[i].y);

ans += a[i].l;

num--;

}

}

if (num)

cout << "No Answer" << endl;

else

cout << ans << endl;

}