1, snt: 2 check: normal check & sang Eratosthenes

void sang() {

fill(prime, prime + limit, true);

prime[0] = prime[1] = false;

for (int i = 2; i <= sqrt(limit); i++) {

if (prime[i]) {

for (int j = i \* i; j <= limit; j += i) prime[j] = false;

}

}

}

bool laSNT(ll n) {

if (n <= limit) return prime[n];

if (n < 2) return false;

if (n < 4) return true;

if (n % 2 == 0 || n % 3 == 0) return false;

for (int i = 5; i <= sqrt(n); i += 6) {

if (n % i == 0 || n % (i + 2) == 0) return false;

}

return true;

}

Nhan xet:

Snt^2 len la so co 3 uoc

a \* b = c khi do c co 4 uoc tro len (a != b, a & b la snt)

p^4 co 5 uoc (p la snt)

trong bai toan phan tich 1 so thanh cac thua so nt thi ta duoc bieu thuc:  
N = a^x \* b ^ y => so luong uoc cua N la: (x + 1) \* (y + 1)

2, so chinh Phuong: la so duoc tao boi n^2;

Nhan xet: 4 la scp co {1, 2, 4}, 9 la scp co {1, 3, 9} => so cac uoc cua 1 scp % 2 != 0;

Cach tinh: 1^2+2^2+3^2+⋯+n^2=n(n+1)(2n+1)​ / 6

Bai: 1 -> N co bao nhieu scp: sqrt(N);

3, uoc so: 2check demuoc & sangboi:

ll solve(ll n) {

int dem = 0;

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

if (n % i == 0) {

dem++;

if (n / i != i) dem++;

}

}

return dem;

}  
  
void sangboi() {

for (ll i = 1; i <= limit; i++) {

for (ll j = i \* i; j <= limit; j += i) {

boi[j]++;

}

}

}

12 => {1, 2, 3, 4, 6, 12} =>6

void sangboi() {

for (ll i = 1; i <= limit; i++) {

for (ll j = 2 \* i; j <= limit; j += i) {

boi[j]++;

}

}

}

12 => {1, 2, 3, 4, 6} =>5

1, tim so luong so chia het cho x tu khoang l -> r:

r/x – (l – 1) / x

2, tim so luong so chia het cho a va b tu 1 -> m:

m / uc(a, b)

3, tim so luong so chia het cho a hoac b tu 1 -> m:

X = m / a;

Y = m / b;

Z = m / uc(a, b);

Cout << x + m – z;

3, hinh hoc: A = (xa, ya), B = (xb, yb);

KCachA-B: sqrt(pow((xa - xb), 2) + pow((ya - yb), 2));

P = (a + b + c) / 2

S = sqrt(p(p-a)(p - b)(p - c));

4, binary\_search:

ll np(ll n, const ll a, ll x) {

l = 1, r = n;

ans = -1;

while(l <= r) {

mid = (l + r) /2;

if (a[mid] == x) return mid;

else if(a[mid] < x) l = mid + 1;

else r = mid – 1;  
}

return ans;  
}

t1 = lower\_bound();

t2 = upper\_bound();

so lan xuat hien cua x: t2 – t1;

vi tri xuat hien dau tien: t1;

vi tri xuat hien cuoi cung: t2-1;

so luong phan tu < x: t1 – 1;

phan tan lon nhat < x: a[t1 - 1];

so luong phan tu > x: n – t2 + 1;

so luong phan tu >= x: n – t1;

5, 2 con tro:

Dang1: 2 con tro 2 dau:

Sort();

L = 1, r = n;

While(l <= r) {

Tu duy ma lam;

}

Dang 2: 2 con tro 1 dau:

Bai 1: dem cac doan con co tong = k(day duong);

Tong = 0;

Int l = 1;

For (int r = 1; r <= n; r++) {

Tong += a[r];

While(tong >= k) {

If (tong == k) dem++;

Tong -= a[l];

L++;

}  
}

Bai 2: dem cac doan con co tong = k(day am);

Unordered\_map<ll, ll> m;

M[0] = 0;

Tong = 0;

Int l = 1;

For (int r = 1; r <= n; r++) {

Tong += a[r];

If (m.find(tong - k) != m.end()) dem+= m[tong - k];

d[tong]++;  
}

bai 3: tim doan con dai nhat co tong = k(day am)

Unordered\_map<ll, ll> m;

M[0] = 0;

Tong = 0;

Int l = 1;

For (int r = 1; r <= n; r++) {

Tong += a[r];

If (m.find(tong - k) != m.end()) dodai = max(dodai, i - m[tong - k]);

If (m.find(tong) == m.end()) m[tong] = I;  
}

M[0] = 0;

K = 3

1 2 3 4 5

1 3 6 10 15

3 – 3 = 0 kiem tra thay m[0] = 0 => dodai = I – m[tong - k] => dodai = 2 – 0 => 2;

bai 4: tim doan con ngan nhat co tong = k(day am)

Unordered\_map<ll, ll> m;

M[0] = 0;

Tong = 0;

Int l = 1;

For (int r = 1; r <= n; r++) {

Tong += a[r];

If (m.find(tong - k) != m.end()) dodai = min(dodai, i - m[tong - k]);

m[tong] = I;  
}