

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tên file	Tên file input	Tên file output	Điểm
1	Cấp số nhân	MULTIPLE.*	MULTIPLE.INP	MULTIPLE.OUT	4
2	Cặp số anh em	PAIRPBRO.*	PAIRPBRO.INP	PAIRPBRO.OUT	5
3	Đua xe địa hình	TERACING.*	TERACING.INP	TERACING.OUT	6
4	Con đường hoa	PAFOWER.*	PAFOWER.INP	PAFOWER.OUT	5

Dấu * là PY hoặc CPP tùy theo ngôn ngữ lập trình là PYTHON hay C++

BÀI 1. CẤP SỐ NHÂN

* Thuật toán đề xuất:

Duyệt giá trị i từ 1 đến n rồi tính lũy thừa và cộng vào kết quả, tuy nhiên tính lũy thừa đến đâu thì phải chia lấy dư cho 2023 ngay lúc đó, tương tự cộng lũy thừa vào kết quả cũng cần phải chia dư cho 2023 ngay thời điểm đó.

Chủ đề: *Duyệt*

* Chương trình minh họa bằng ngôn ngữ lập trình C++:

```
#include <bits/stdc++.h>
#define base 2023
using namespace std;
int n,q,ans=1,lt=1;
int main()
{
    freopen("multiple.inp","r",stdin);
    freopen("multiple.out","w",stdout);
    cin >> n >> q;
    for (int i=1; i<n; i++)
    {
        lt = (lt * q) % base;
        ans = (ans + lt) % base;
    }
    cout << ans;
    return 0;
}
```

* Chương trình minh họa bằng ngôn ngữ lập trình Python:

```
import sys
sys.stdin = open("multiple.inp","r")
```

```

sys.stdout = open("multiple.out", "w")
n, q = map(int, input().split())
ans = 1;
lt = 1;
for i in range(1, n):
    lt = (lt * q) % 2023;
    ans = (ans + lt) % 2023;
print(ans)

```

BÀI 2. CẬP SỐ ANH EM

*** Thuật toán đề xuất:**

Subtask 1 (30% số test) $1 < n \leq 10^2$:

Duyệt 2 giá trị x và y ($1 \leq x < y \leq n$), với hai số nguyên tố x, y nếu $y - x = k$ thì tăng biến đếm kết quả 1 đơn vị.

Độ phức tạp: $O(n^2)$

Subtask 2 (70% số test) $10^2 < n \leq 10^6$:

Sử dụng sàng nguyên tố tạo mảng $p[]$ với ý nghĩa:

- $p[i] = 1$ nếu i là số nguyên tố và $p[i] = 0$ trong trường hợp ngược lại.

Duyệt $x: 1 \rightarrow n - k$:

Nếu $p[x] == 1$ và $p[x + k] == 1$ thì x và $x + k$ là những số nguyên tố khi đó tăng biến kết quả 1 đơn vị.

Độ phức tạp:

- Sàng số nguyên tố $O(n \log n)$
- Duyệt tìm kết quả $O(n)$

Độ phức tạp cuối cùng: $O(n \log n)$

Chủ đề: duyệt + số học

*** Chương trình minh họa bằng ngôn ngữ lập trình C++:**

```

#include <bits/stdc++.h>
#define N 1000000
using namespace std;
int p[N+3];
int n, k, ans;
void sangngto(int n)
{
    fill(p+2, p+n+1, 1);
    for (int i=2; i*i <= n; i++)
        if (p[i] == 1)
            for (int j=i*i; j<=n; j+=i)
                p[j] = 0;
}

```

```

}
int main()
{
    freopen("pairpbro.inp","r",stdin);
    freopen("pairpbro.out","w",stdout);
    cin >> n >> k;
    sangngto(n);
    for (int x = 2; x + k <= n; x++)
        if (p[x] == 1 && p[x+k] == 1) ans++;
    cout << ans;
    return 0;
}

```

*** Chương trình minh họa bằng ngôn ngữ lập trình Python:**

```

N = int(1e6)
def sangngto():
    c[0] = c[1] = 0;
    k = int(N ** 0.5)
    for i in range(2, k+1):
        if (c[i] == 1):
            for j in range(i*i, N+1,i): c[j] = 0
import sys
sys.stdin = open("pairpbro.inp","r")
sys.stdout = open("pairpbro.out","w")
c = [1] * (N + 3);
sangngto()
n, k = map(int,input().split())
ans = 0
for x in range(2, n - k + 1):
    if (c[x] == 1 and c[x + k] == 1):
        ans += 1
print(ans)

```

BÀI 3. ĐUA XE ĐỊA HÌNH

*** Thuật toán đề xuất:**

Subtask 1 (60% số test) $1 < n \leq 10^3; 2 \leq k \leq 10^3$;

Duyệt cặp (i, j) : $1 \leq i < j \leq n$:

Nếu $|a[j] - a[i]| = k$ thì tăng kết quả 1 đơn vị

Độ phức tạp: $O(n^2)$

Subtask 2 (20% số test) $10^3 < n \leq 10^4; 2 \leq k \leq 10^5$;

Sắp xếp dãy a tăng dần.

Bài toán trở thành: đếm cặp (i, j) (với $1 \leq i < j \leq n$) sao cho $a[j] - a[i] = k \Leftrightarrow a[i] = a[j] - k$.

Chuẩn bị trước mảng $c[x]$ là tần số của giá trị x ở trong dãy a .

Duyệt các giá trị $a[i]$: cần đếm phía trước $a[i]$ có bao nhiêu giá trị bằng $a[i] - k$, việc đếm này thực hiện nhanh nhờ mảng c với lời gọi đến $c[a[i] - k]$.

Subtask 3 (20% số test) $10^4 < n \leq 10^5$; $2 \leq k \leq 10^9$.

Tương tự cách làm ở **Subtask 2**, tuy nhiên để đếm tần số xuất hiện của các giá trị trong a thì chúng ta sẽ sử dụng cấu trúc dữ liệu **map** (trong C++) hoặc kiểu từ điển (trong Python) thay vì dùng mảng.

Chủ đề: Duyệt

*** Chương trình minh họa bằng ngôn ngữ lập trình C++:**

```
#include <bits/stdc++.h>
#define N 100000
#define ll long long
using namespace std;
int a[N+3];
int n, k;
ll ans;
map <int,int> M;
int main()
{
    freopen("teracing.inp","r",stdin);
    freopen("teracing.out","w",stdout);
    cin >> n >> k;
    for (int i=1; i<=n; i++) cin >> a[i];
    sort(a+1,a+n+1);
    for (int i=1; i<=n; i++)
    {
        if (a[i] > k) ans += M[a[i] - k];
        M[a[i]]++;
    }
    cout << ans;
    return 0;
}
```

*** Chương trình minh họa bằng ngôn ngữ lập trình Python:**

```
import sys
sys.stdin = open("teracing.inp","r")
sys.stdout = open("teracing.out","w")
n, k = map(int,input().split())
a = list(map(int,input().split()))
a.sort()
dic = {a[0]:1}
ans = 0;
for i in range(1,n):
    if (a[i] > k and dic.get(a[i] - k) != None):
        ans += dic[a[i]-k];
```



```

if (dic.get(a[i]) == None): dic[a[i]] = 1
else: dic[a[i]] += 1;
print(ans)

```

BÀI 4. CON ĐƯỜNG HOA

* Thuật toán đề xuất:

Gọi $dp[i][j]$ là số lượng cặp cây hoa được giữ lại nhiều nhất khi xét i cây hoa đầu tiên bên lề A và j cây hoa đầu tiên bên lề B.

Khởi tạo:

- $dp[i][0] = 0$ – nếu bên lề B không có cây hoa nào thì số lượng cặp cây hoa cần giữ lại bằng 0.
- $dp[0][j] = 0$ – nếu bên lề A không có cây hoa nào thì số lượng cặp cây hoa cần giữ lại bằng 0.

Xét cây hoa thứ i bên lề A và cây hoa thứ j bên lề B:

- Nếu $a[i] = b[j] = -1$: cần loại bỏ đi 1 trong hai cây hoa ở lề A hoặc lề B
 - Nếu bỏ đi cây hoa thứ i ở lề A thì: $dp[i][j] = dp[i-1][j]$
 - Nếu bỏ đi cây hoa thứ j ở lề B thì: $dp[i][j] = dp[i][j-1]$

Vì cần số lượng cặp cây hoa là lớn nhất nên:

$$dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$$

- Ngược lại: tức là sẽ giữ lại cả cây hoa thứ i bên lề A và cây hoa thứ j bên lề B nên:

$$dp[i][j] = dp[i-1][j-1] + 1$$

Tóm lại công thức quy hoạch động như sau:

$$\begin{cases} \text{Nếu } a[i] = b[j] = -1: dp[i][j] = \max(dp[i-1][j], dp[i][j-1]) \\ \text{Ngược lại: } dp[i][j] = dp[i-1][j-1] + 1 \end{cases}$$

Kết quả bài toán: $dp[n][n]$

Độ phức tạp: $O(n^2)$

Chủ đề: *Quy hoạch động*

* Chương trình minh họa bằng ngôn ngữ lập trình C++:

```

#include <bits/stdc++.h>
#define N 1000
using namespace std;
int dp[N+3][N+3], a[N+3], b[N+3];
int n;
int main()
{

```

```
freopen("paflower.inp", "r", stdin);
freopen("paflower.out", "w", stdout);
cin >> n;
for (int i=1; i<=n; i++)
    cin >> a[i] >> b[i];
for (int i = 1; i <= n; i++)
    for (int j = 1; j <= n; j++)
        if (a[i] == b[j] && a[i] == -1)
            dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
        else
            dp[i][j] = dp[i-1][j-1] + 1;
cout << dp[n][n];
return 0;
}
```
