SỞ GIÁO DỤC VÀ ĐÀO TẠO
TỈNH QUẢNG NINH

ĐỀ THI MINH HỌA

KỲ THI CHỌN HỌC SINH GIỎI CẤP TỈNH THCS
Môn thi: **TIN HỌC** - Bảng **B**

*Thời gian làm bài: **150 phút**, không kể thời gian phát đề*

(Đề thi này có 03 trang)

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tệp chương trình	Tệp dữ liệu	Tệp kết quả	Bộ nhớ (MB)	Thời gian (giây)	Điểm
1	Căn bậc hai	`rgc.*`	`rgc.inp`	`rgc.out`	1024	1	6
2	Bóng đá	`bal.*`	`bal.inp`	`bal.out`	1024	1	6
3	Số lớn nhất	`max.*`	`max.inp`	`max.out`	1024	1	5
4	Chia mảng	`arr.*`	`arr.inp`	`arr.out`	1024	1	3

Dấu `*` được thay thế bởi `cpp` hoặc `py` của ngôn ngữ lập trình được sử dụng tương ứng là `C++` hoặc `Python`.

Hãy lập trình giải các bài toán sau:

Bài 1. Căn bậc hai (6 điểm)

Hôm nay, lớp của An làm bài tập môn Toán về nội dung đưa thừa số ra ngoài dấu căn. Cô giáo giao bài tập về nhà là cho trước số nguyên dương n, hãy tìm hai số nguyên dương x, y sao cho $\sqrt{n} = x \times \sqrt{y}$ với x lớn nhất.

Ví dụ:

- Với $n = 18$: do $\sqrt{18} = 3 \times \sqrt{2}$ nên $x = 3, y = 2$;
- Với $n = 7$: do $\sqrt{7} = 1 \times \sqrt{7}$ nên $x = 1, y = 7$;
- Với $n = 4$: do $\sqrt{4} = 2 \times \sqrt{1}$ nên $x = 2, y = 1$.

Dữ liệu: Vào từ tệp văn bản `rgc.inp` gồm một dòng chứa số nguyên n ($1 \le n \le 10^{14}$).

Kết quả: Ghi ra tệp văn bản `rgc.out` gồm một dòng chứa hai số nguyên x và y.

Ví dụ:

rgc.inp	rgc.out
18	3 2
7	1 7
4	2 1

Ràng buộc:

- Có 30% số test ứng với 30% số điểm thỏa mãn: $1 \le n \le 10^5$;
- 30% số test khác ứng với 30% số điểm thỏa mãn: $1 \le n \le 10^9$;
- 40% số test còn lại ứng với 40% số điểm: Không có thêm ràng buộc nào.

Bài 2. Bóng đá (6 điểm)

Trong một giải đấu bóng đá có n đội bóng tham gia. Mỗi đội bóng phải đá với mỗi đội bóng khác đúng 2 trận lượt đi và về, trong đó có 1 trận trên sân nhà và 1 trận trên sân khách.

Mỗi đội bóng i có 2 trang phục thi đấu, 1 trang phục dành cho sân nhà có màu được biểu diễn bởi số nguyên x_i, còn 1 trang phục dành cho sân khách có màu được biểu diễn bởi số nguyên y_i ($x_i \ne y_i$).

Các trang phục cùng màu thì được biểu diễn bởi cùng một số và các trang phục khác màu thì được biểu diễn bởi các số khác nhau.

Trong mỗi trận đấu, một đội mặc trang phục sân nhà, một đội mặc trang phục sân khách. Tuy nhiên, nếu hai đội cùng màu trang phục thì đội khách được phép thay trang phục sân nhà của mình.

Hãy tính xem trong giải đấu, mỗi đội mặc bao nhiêu lần trang phục sân nhà và bao nhiêu lần trang phục sân khách.

Dữ liệu: Vào từ tệp văn bản `bal.inp`. Dòng đầu tiên chứa số nguyên n $(2 \le n \le 10^5)$. Dòng thứ i trong n dòng tiếp theo chứa hai số nguyên x_i, y_i $(1 \le x_i, y_i \le 10^9; x_i \ne y_i)$.

Kết quả: Ghi ra tệp văn bản `bal.out` gồm n dòng, trong đó dòng thứ i chứa hai số nguyên tương ứng là số lần mặc trang phục sân nhà và sân khách của đội bóng i.

Ví dụ:

bal.inp	**bal.out**
3 4 7 7 4 4 2	3 1 4 0 2 2

Ràng buộc:

- Có 40% số test ứng với 40% số điểm thỏa mãn: $2 \le n \le 1000$;
- 30% số test khác ứng với 30% số điểm thỏa mãn: $1 \le x_i, y_i \le 10^5$;
- 30% số test còn lại ứng với 30% số điểm: Không có thêm ràng buộc nào.

Bài 3. Số lớn nhất (5 điểm)

Cho hai số nguyên dương a và b. Hãy tìm số c lớn nhất chứa tất cả các chữ số của a và b, trong đó thứ tự xuất hiện các chữ số của a và b được giữ nguyên.

Dữ liệu: Vào từ tệp văn bản `max.inp`. Dòng đầu tiên chứa số nguyên dương a và dòng thứ hai chứa số nguyên dương b (a, b có không quá 1000 chữ số).

Kết quả: Ghi ra tệp văn bản `max.out` gồm một dòng chứa số c tìm được.

Ví dụ:

max.inp	**max.out**
20 4181	421810

Ràng buộc:

- Có 40% số test ứng với 40% số điểm thỏa mãn: a có 1 chữ số;
- 30% số test khác ứng với 30% số điểm thỏa mãn: Không có chữ số nào của a xuất hiện trong b;
- 30% số test còn lại ứng với 30% số điểm: Không có thêm ràng buộc nào.

Bài 4. Chia mảng (3 điểm)

Cho một mảng gồm n số nguyên dương. Nhiệm vụ của bạn là chia nó thành k đoạn, sao cho tổng lớn nhất trên một đoạn là nhỏ nhất có thể.

Dữ liệu: Dòng đầu tiên chứa hai số nguyên n và k $(1 \le k \le n \le 10^5)$. Dòng thứ hai chứa các phần tử a_i của mảng $(1 \le a_i \le 10^9)$.

Kết quả: Ghi ra tổng lớn nhất trên một đoạn là nhỏ nhất.

Ví dụ:

arr.inp	**arr.out**

10 4 1 3 2 4 10 8 4 2 5 3	12

Ràng buộc:

- Có 10% số test ứng với 10% số điểm thỏa mãn: $k = 1$;
- 10% số test khác ứng với 10% số điểm thỏa mãn: $k = 2$;
- 40% số test khác ứng với 40% số điểm thỏa mãn: $1 \leq n \leq 10^2, 1 \leq a_i \leq 10^3$;
- 40% số test còn lại ứng với 40% số điểm: Không có thêm ràng buộc nào.

---------- HẾT ----------

- *Thí sinh không được sử dụng tài liệu;*
- *Giám thị không giải thích gì thêm.*

Họ và tên thí sinh: .....Số báo danh: .....

SỞ GIÁO DỤC VÀ ĐÀO TẠO
TỈNH QUẢNG NINH

ĐỀ THI MINH HỌA

HƯỚNG DẪN CHẤM THI CHỌN HỌC SINH GIỎI CẤP TỈNH THCS
Môn thi: **TIN HỌC** - Bảng **B**

(*Hướng dẫn này có 06 trang*)

I. HƯỚNG DẪN CHUNG

1. Giám khảo nghiên cứu đề, hướng dẫn chấm, test, code;
2. Chép bài của thí sinh vào máy chấm. Kiểm tra bài thí sinh trên giấy in và trên máy đảm bảo trùng khớp;
3. Thực hiện chấm bài trên máy chấm (bài thi của thí sinh được chấm trên máy tính bằng phần mềm chấm thi Themis, bản quyền của Cục Quản lý chất lượng);
4. Kiểm tra kết quả chấm bài (kiểm tra lại các bài có kết quả như không tồn tại bài, lỗi biên dịch, các bài bị 0 điểm, …);
5. Tổng hợp kết quả chấm bài của thí sinh. Điểm bài thi được xuất từ phần mềm chấm thi, không quy tròn điểm của từng câu, điểm của bài thi;
6. Sau khi chấm bài xong, giám khảo lưu lại toàn bộ dữ liệu của chấm thi bằng phần mềm chấm thi Themis (Kỳ thi → Ghi kỳ thi), ghi ra đĩa CD và nộp cho hội đồng chấm thi để lưu.

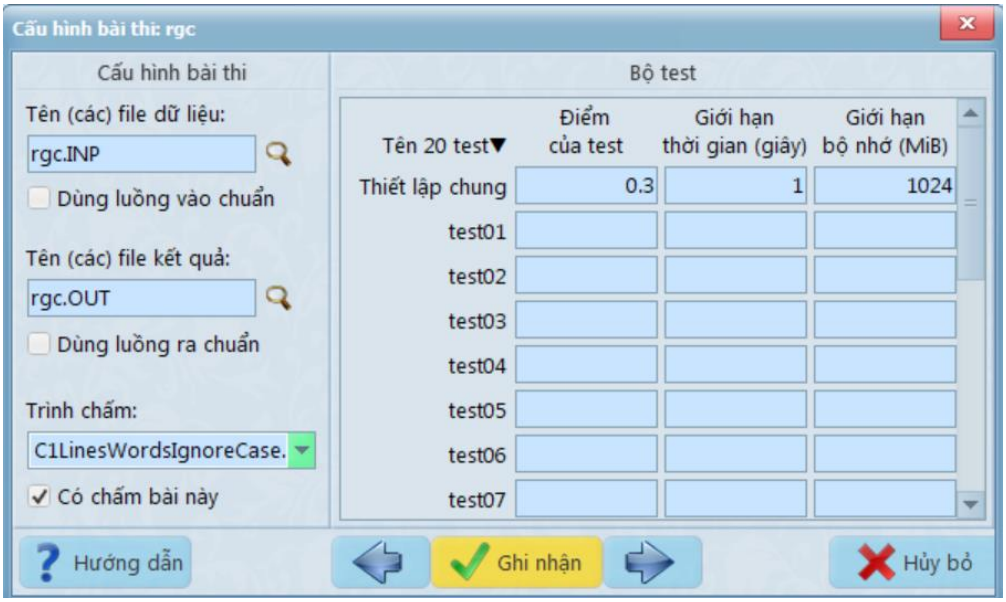
II. ĐÁP ÁN, BIỂU ĐIỂM

Bài 1. Căn bậc hai (6 điểm)

Phân bổ điểm

- Có 30% số test ứng với 30% số điểm thỏa mãn: $1 \le n \le 10^5$;
- 30% số test khác ứng với 30% số điểm thỏa mãn: $1 \le n \le 10^9$;
- 40% số test còn lại ứng với 40% số điểm: Không có thêm ràng buộc nào.

Cấu hình bài thi



Hướng dẫn giải

Subtask 1 (30%): $1 \le n \le 10^5$

Ta có $\sqrt{n} = x\sqrt{y} \Leftrightarrow n = x^2.y$.

Duyệt các giá trị y từ 1 đến n và kiểm tra n có chia hết cho y hay không và $\frac{n}{y}$ có là số chính phương hay không. Nếu có thì ta tính $x = \sqrt{\frac{n}{y}}$.

Vì ta cần tìm x lớn nhất, tức là y nhỏ nhất, nên khi tìm được cặp (x, y) đầu tiên ta sẽ ghi ra kết quả và

kết thúc chương trình.

Độ phức tạp thuật toán $O(n)$.

Subtask 2 (30%): $1 \le n \le 10^9$

Nhận thấy rằng $1 \le x \le \sqrt{n}$ và cần tìm x lớn nhất, do đó ta chỉ cần duyệt các giá trị x từ $\lfloor \sqrt{n} \rfloor$ về 1. Với mỗi giá trị x, ta cần kiểm tra n có chia hết cho x^2 hay không. Nếu có thì ta sẽ tính $y = \frac{n}{x^2}$. Khi tìm được cặp (x, y) đầu tiên ta sẽ ghi ra kết quả và kết thúc chương trình.

Trong subtask này, x và y chỉ dùng kiểu số nguyên int.

Độ phức tạp thuật toán $O(\sqrt{n})$.

Subtask 3 (40%): Không có thêm ràng buộc nào

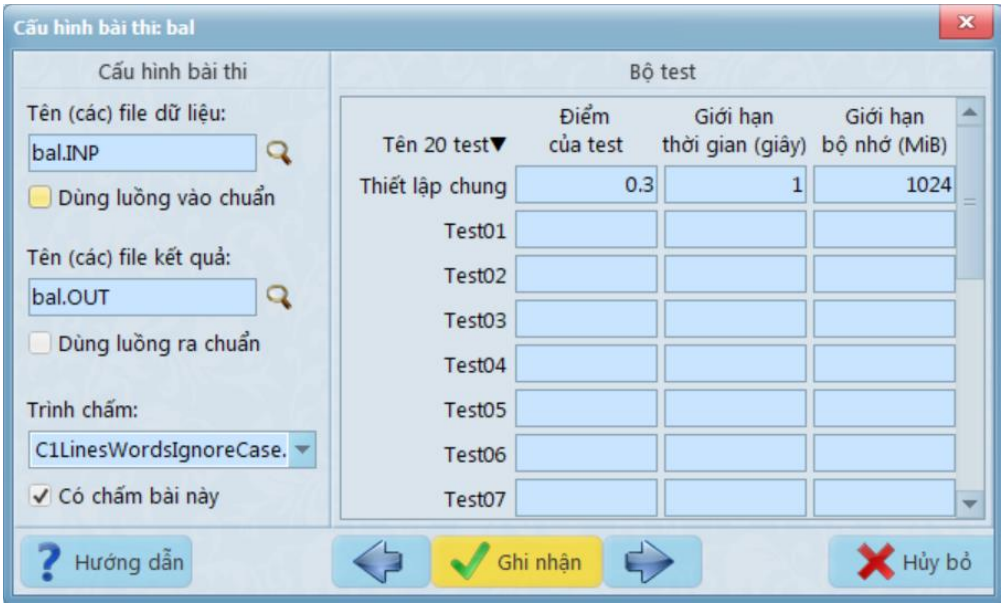
Thuật toán trong subtask 3 này giống với subtask 2, nhưng cần khai báo x, y kiểu số nguyên $long\ long$.

Bài 2. Bóng đá (6 điểm)

Phân bổ điểm

- Có 40% số test ứng với 40% số điểm thỏa mãn: $2 \le n \le 1000$;
- 30% số test khác ứng với 30% số điểm thỏa mãn: $1 \le x_i, y_i \le 10^5$;
- 30% số test còn lại ứng với 30% số điểm: Không có thêm ràng buộc nào.

Cấu hình bài thi



Hướng dẫn giải

Subtask 1 (40%): $2 \le n \le 1000$

Gọi $cnt1[i], cnt2[i]$ lần lượt là số lần mặc trang phục sân nhà, sân khách của đội i $(i = 1, 2, \dots, n)$. Ban đầu ta khởi tạo $cnt1[i] = cnt2[i] = 0$.

Với mỗi trận đấu giữa đội nhà i và đội khách j $(i = 1, 2, \dots, n; j = 1, 2, \dots, n; i \ne j)$:

- Đội i luôn mặc trang phục sân nhà. Do đó tăng số lần mặc trang phục sân nhà của đội i lên 1, tức là tăng $cnt1[i]$ lên 1;
- Kiểm tra đội j mặc trang phục sân khách hay sân nhà:
  - Nếu $y[j] \ne x[i]$ thì đội j cần mặc trang phục sân khách, tức là tăng $cnt2[j]$ lên 1.
  - Nếu $y[j] = x[i]$ thì đội j cần mặc trang phục sân nhà, tức là $cnt1[j]$ lên 1.

Độ phức tạp thuật toán là $O(n^2)$.

Subtask 2 (30%): $1 \le x_i, y_i \le 10^5$

Đầu tiên ta nhận thấy rằng mỗi đội i sẽ thi đấu $(n-1)$ trận sân nhà. Trong các trận này, đội i sẽ mặc trang phục sân nhà.

Mặt khác đội i sẽ thi đấu $(n-1)$ trận sân khách với các đội bóng j còn lại. Nếu trang phục sân khách của đội i trùng với trang phục sân nhà của đội j thì đội i sẽ mặc trang phục sân nhà. Do đó nếu ta gọi $cnt[c]$ là số đội bóng có trang phục sân nhà màu c thì số lần đội i mặc trang phục sân nhà khi thi đấu trên sân khách là $cnt[y[i]]$. Mảng cnt này cần được tính toán từ trước.

Vậy tổng số lần đội i mặc trang phục sân nhà là $(n-1) + cnt[y[i]]$.

Vì mỗi đội sẽ thi đấu $2(n-1)$ trận nên số lần mặc trang phục sân khách của đội i là:

$$2(n-1) - \Big((n-1) + cnt[y[i]]\Big) = (n-1) - cnt[y[i]]$$

Trong subtask này, do $1 \le x_i, y_i \le 10^5$ nên ta có thể dùng biến cnt kiểu mảng.

Độ phức tạp thuật toán là $O(n)$.

Subtask 3 (30%): Không có thêm ràng buộc nào

Thuật toán trong subtask 3 này giống như subtask 2, nhưng có một điểm khác là do $1 \le x_i, y_i \le 10^9$ nên ta không thể dùng biến cnt kiểu mảng mà cần dùng kiểu map thay thế.

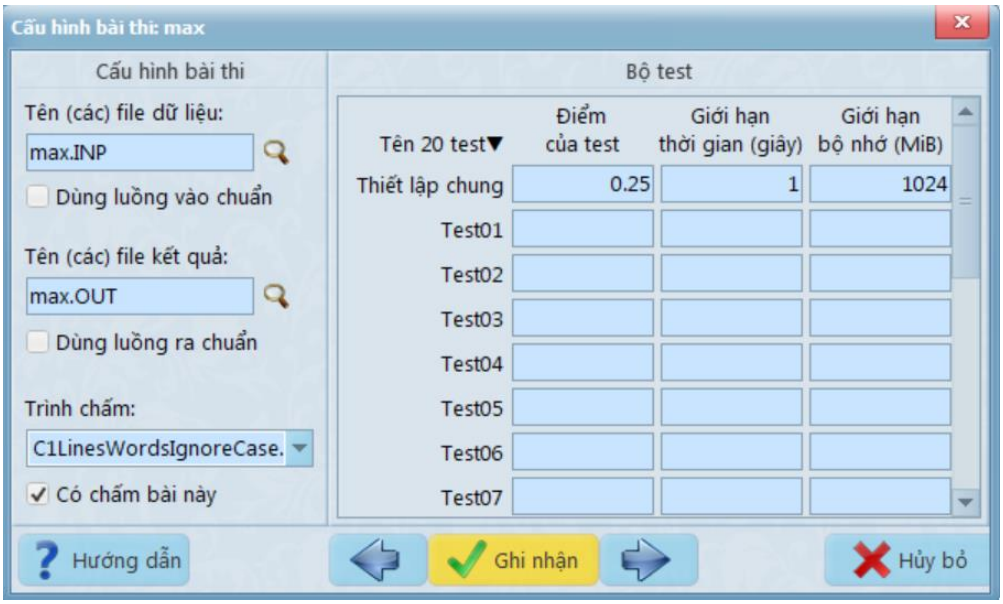
Độ phức tạp thuật toán là $O(n)$.

Bài 3. Số lớn nhất (5 điểm)

Phân bổ điểm

- Có 40% số test ứng với 40% số điểm thỏa mãn: a có 1 chữ số;
- 30% số test khác ứng với 30% số điểm thỏa mãn: Không có chữ số nào của a xuất hiện trong b;
- 30% số test còn lại ứng với 30% số điểm: Không có thêm ràng buộc nào.

Cấu hình bài thi



Hướng dẫn giải

Do các số a, b có tối đa 10^5 chữ số nên ta cần lưu trữ các số a, b bằng kiểu xâu.

Gọi m, n lần lượt là số chữ số của a, b.

Subtask 1 (40%): a có 1 chữ số

Ta tìm chữ số đầu tiên trong số b mà nhỏ hơn chữ số duy nhất của a. Sau đó chèn chữ số của số a vào vị trí này trong số b. Trường hợp nếu không có chữ số nào trong b nhỏ hơn chữ số của a thì ta chèn chữ số của a vào cuối b.

Độ phức tạp thuật toán là $O(n)$.

Subtask 2 (30%): Không có chữ số nào của a xuất hiện trong b

Ta sử dụng hai con trỏ i, j để duyệt qua từng chữ số của số a, b.

Do không có chữ số nào của a xuất hiện trong b nên chỉ xảy ra hai trường hợp sau:

- $a[i] > b[j]$: Thêm chữ số $a[i]$ vào cuối số c hiện tại và tăng con trỏ i lên 1;
- $a[i] < b[j]$: Thêm chữ số $b[j]$ vào cuối số c hiện tại và tăng con trỏ j lên 1;

Việc duyệt trên sẽ dừng khi i hoặc j đến chữ số cuối cùng của a và b. Cuối cùng ta thêm các chữ số còn lại chưa được xét của a hoặc b vào cuối số c.

```
string a, b, c = "";
cin >> a >> b;
int m = a.size(), n = b.size(), i = 0, j = 0;
while (i < m && j < n) {
    if (a[i] > b[j])
        c += a[i++];
    else
        c += b[j++];
}
while (i < m)
    c += a[i++];
while (j < n)
    c += b[j++];
cout << c << "\n";
```

Độ phức tạp thuật toán là $O(m + n)$.

Subtask 3 (30%): Không có thêm ràng buộc nào

Ta sử thuật toán đã trình bày trong subtask 2 nhưng bây giờ phải xét thêm trường hợp khi $a[i] = b[j]$. Khi đó ta cần tiếp tục so sánh các chữ số tiếp theo cho đến khi tìm thấy chữ số đầu tiên của a và b khác nhau hoặc đến khi một trong hai xâu kết thúc. Nếu một trong hai xâu có chữ số lớn hơn, thì thêm chữ số ở xâu đó vào kết quả. Ta có thể thay thế việc tìm kiếm này bằng so sánh xâu con của a từ vị trí i đến hết xâu và xâu con của b từ vị trí j đến hết xâu.

```
string a, b, c = "";
cin >> a >> b;
int m = a.size(), n = b.size(), i = 0, j = 0;
while (i < m && j < n) {
    if (a.substr(i) >= b.substr(j))
        c += a[i++];
    else
        c += b[j++];
}
while (i < m)
    c += a[i++];
while (j < n)
    c += b[j++];
cout << c << "\n";
```

Độ phức tạp thuật toán là $O(m^2 + n^2)$.

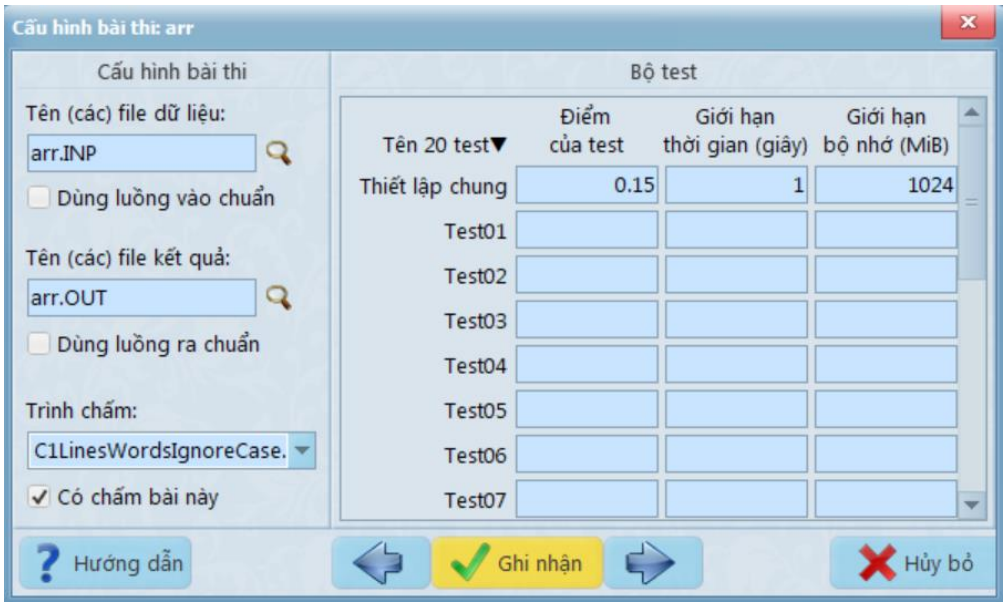
Bài 4. Chia mảng (3 điểm)

Phân bổ điểm

- Có 10% số test ứng với 10% số điểm thỏa mãn: $k = 1$;
- 10% số test khác ứng với 10% số điểm thỏa mãn: $k = 2$;
- 40% số test khác ứng với 40% số điểm thỏa mãn: $1 \le n \le 10^2, 1 \le a_i \le 10^3$;

- 40% số test còn lại ứng với 40% số điểm: Không có thêm ràng buộc nào.

Cấu hình bài thi



Hướng dẫn giải

Subtask 1 (10%): $k = 1$

Trong trường hợp cả mảng a chỉ là một đoạn, vì vậy câu trả lời là $a_1 + a_2 + \cdots + a_n$.

Độ phức tạp thuật toán là $O(n)$.

Subtask 2 (10%): $k = 2$

Trong trường hợp này ta cần chia mảng a thành 2 đoạn $a_1, a_2, \ldots, a_i$ và $a_{i+1}, a_{i+2}, \ldots, a_n$ sao cho $\max(a_1 + a_2 + \cdots + a_i, a_{i+1} + a_{i+2} + \cdots + a_n)$ là nhỏ nhất.

Ta duyệt các giá trị $i = 1, 2, \ldots, n-1$. Với mỗi i, ta cần tính:

$$\max(a_1 + a_2 + \cdots + a_i, a_{i+1} + a_{i+2} + \cdots + a_n) = \max(pref_i, sum - pref_i)$$

và cập nhật giá trị max nhỏ nhất cùng chỉ số i tương ứng, trong đó $pref_i = a_1 + a_2 + \cdots + a_i$ và $sum = a_1 + a_2 + \cdots + a_n$.

Việc tính sum được thực hiện trước khi duyệt i, còn việc tính $pref_i$ cũng có thể thực hiện trước hoặc kế thừa trong quá trình tính ở bước trước.

Độ phức tạp thuật toán là $O(n)$.

Subtask 3 (40%): $1 \le n \le 10^2, 1 \le a_i \le 10^3$

Trước tiên ta xét bài toán kiểm tra xem nếu các đoạn có tổng không quá t thì có thể chia mảng thành k đoạn được không. Hàm $check(t)$ sau đây thực hiện công việc đó.

```
bool check(long long t) {
    long long s = 0, num = 1;
    for (int i = 1; i <= n; i++) {
        if (a[i] > t) return false;
        if (s + a[i] <= t)
            s += a[i];
        else {
            num++;
            s = a[i];
        }
    }
    return (num <= k);
}
```

Bây giờ ta sẽ sử dụng hàm $check(t)$ trên để tìm câu trả lời của bài toán. Ta sẽ xét các giá trị t lần lượt từ 1 trở đi. Nếu với giá trị t hiện tại mà ta có thể chia mảng thành k đoạn thì ta đưa ra câu trả lời là t và kết thúc. Ngược lại ta sẽ tăng t lên 1.

Độ phức tạp thuật toán là $O\big(n \times (a_1 + a_2 + \cdots + a_i)\big)$.

Subtask 4 (40%): Không có thêm ràng buộc nào

Ta sẽ cải tiến thuật toán trong subtask 3 dựa trên nhận xét sau. Nếu với một giá trị $t = c$ mà ta có thể chia mảng thành k đoạn thì các giá trị $t < c$ ta cũng có thể chia mảng thành k đoạn. Còn nếu một giá trị $t = c$ mà ta không thể chia mảng thành k đoạn thì các giá trị $t > c$ ta cũng không thể chia mảng thành k đoạn.

Dựa vào nhận xét trên, ta có thể tìm giá trị t nhỏ nhất thỏa mãn bằng thuật toán tìm kiếm nhị phân.

```
long long l = 0, r = 1e18;
while (l < r) {
    long long c = (l + r) / 2;
    if (check(c))
        r = c;
    else
        l = c + 1;
}
cout << l << '\n';
```

Độ phức tạp thuật toán là $O\big(n \times \log_2(a_1 + a_2 + \cdots + a_n)\big)$

---------- HẾT ----------