



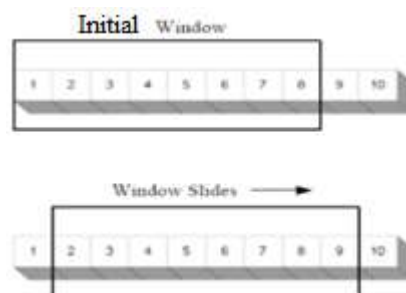
BÀI GIẢNG VỀ KỸ THUẬT LẬP TRÌNH VỚI C++

05 KỸ THUẬT DỊCH CHUYỂN CỬA SỐ – SLIDING WINDOW TECHNIQUE



I. Ý tưởng

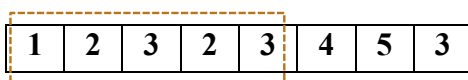
Kỹ thuật “**dịch chuyển cửa sổ**” thường được áp dụng cho các bài toán xử lý “**các phần tử liên tiếp**” trên dãy hoặc bảng.



Trong kỹ thuật dịch chuyển cửa sổ, ta duy trì một số các phần tử liên tiếp nằm trong cửa sổ và lưu thông tin đã xử lý của các phần tử này. Quá trình dịch cửa sổ sẽ thêm vào một số phần tử và lấy ra một số phần tử (ở đầu khác) ra khỏi cửa sổ, việc xử lý thông tin các phần tử trong cửa sổ lúc này **không xử lý lại hoàn toàn** mà chỉ xử lý **các phần tử được thêm vào** và **các phần tử bị lấy ra khỏi cửa sổ**.

Ví dụ: Cho dãy gồm n số a_1, a_2, \dots, a_n và số k , tìm k số hạng liên tiếp có tổng lớn nhất.

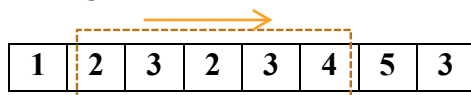
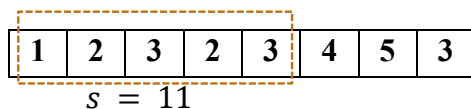
Giải: Ví dụ $n = 10$ và $k = 5$.



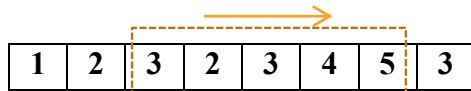
Cửa sổ ban đầu chứa 3 số đầu tiên a_1, a_2, a_3, a_4, a_5 tức là: 1, 2, 3, 2, 3 và tổng bằng $s = 11$.

Từng bước dịch chuyển cửa sổ sang phải, mỗi lần dịch chuyển sẽ thêm vào một phần tử (bên phải cửa sổ hiện tại) và bỏ ra một phần tử (bên trái cửa sổ hiện tại). Việc tính tổng các phần tử trong cửa sổ được cập nhật bằng cách cộng phần tử được thêm vào và trừ phần tử được lấy ra.

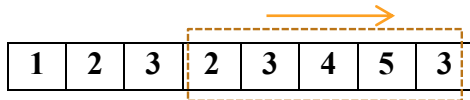
Dịch cửa sổ sang phải ta có:



$$s = s + 4 - 1 = 11 + 4 - 1 = 14$$



$$s = s + 5 - 2 = 14 + 5 - 2 = 17$$



$$s = s + 3 - 3 = 17 + 3 - 3 = 17$$

Vậy kết quả là 17.

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int n, k, a[1000000];
4  long long kq, s = 0;
5  int main(){
6      cin>>n>>k;
7      for(int i = 1; i <= n; ++i) cin>>a[i];
8      //Initial Window
9      for (int i = 1; i <= k; ++i) s += a[i];
10     kq = s;
11     //Window slides -->
12     for(int i = k+1; i<= n; ++i){
13         //update sum window
14         s = s + a[i] - a[i-k];
15         // update result
16         kq = max(kq, s);
17     }
18     cout<<kq;
19 }
```

❖ Chú ý:

Kích thước cửa sổ có thể thay đổi tùy vào yêu cầu của bài toán.

BÀI TẬP



1☀. Odd – Even Window

Cho dãy gồm n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$) và số nguyên k ($2 \leq k \leq n \leq 10^6$). Hãy tìm k phân tử liên tiếp trong dãy sao cho:

- Có ít nhất một số chẵn và một số lẻ;
- Tổng k số hạng này là nhỏ nhất.

Dữ liệu cho trong file OEWIN.INP gồm:

- Dòng đầu ghi hai số nguyên dương n và k .
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$).

Kết quả ghi ra file OEWIN.OUT là tổng nhỏ nhất của k số hạng liên tiếp thỏa mãn đề bài. (Dữ liệu luôn đảm bảo có k số hạng thỏa mãn).

Ví dụ:

OWIN.INP	OWIN.OUT
5 3	12
1 2 9 10 -1	

**2☀. Difference_K**

Cho 2 dãy số nguyên A: a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$) và số B: b_1, b_2, \dots, b_m ($|b_i| \leq 10^9$) và số nguyên k ($1 \leq k \leq \min(n, m)$). Ta gọi $DifferenceK(A, B)$ là $\min |SubA(k) - SubB(k)|$ trong đó $SubA(k)$ là tổng của k phần tử liên tiếp nào đó trong dãy A, $SubB(k)$ là tổng k phần tử liên tiếp nào đó trong dãy B.

Yêu cầu: Tìm $DifferenceK(A, B)$.

Dữ liệu cho trong file *DifferenceK.Inp* gồm:

- Dòng đầu ghi ba số nguyên dương n, m và k ($n, m \leq 10^5$).
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$).
- Dòng thứ hai ghi m số nguyên b_1, b_2, \dots, b_m ($|b_i| \leq 10^9$).

Kết quả ghi ra file *DifferenceK.Out* là giá trị $DifferenceK(A, B)$.

Ví dụ:

<i>DifferenceK.Inp</i>	<i>DifferenceK.Out</i>
4 5 3 1 2 3 4 1 4 2 2 4	1

Giải thích: Chọn $SubA(3) = 1 + 2 + 3 = 6$, $SubB(3) = 1 + 4 + 2 = 7$.

**3☀. Strong Password**

Một mật khẩu được gọi là mạnh nếu độ dài một khẩu đó không nhỏ hơn 6 và chứa đủ ba loại kí tự:

- Kí tự số: '0', '1', .., '9'.
- Kí tự chữ thường: 'a', .., 'z'.
- Kí tự chữ hoa: 'A', .., 'Z'.

Yêu cầu: Cho xâu kí tự *St* gồm các kí tự thuộc ba loại như trên. Hãy tìm một xâu con gồm các kí tự liên tiếp để làm mật khẩu sao cho: là mật khẩu mạnh và độ dài nhỏ nhất có thể.

Dữ liệu cho trong file SPASS.INP gồm xâu *St* có số kí tự không quá 10^5 .

Kết quả ghi ra file SPASS.OUT là độ dài của xâu con gồm các kí tự liên tiếp để làm mật khẩu.

Ví dụ:

SPASS.INP	SPASS.OUT
12aaaabbAnn	8

**4☀. XÂU 26**

Cho xâu *St* chỉ gồm các kí tự thuộc tập 'a' .. 'z'. Hãy tìm một xâu con gồm các kí tự liên tiếp của xâu *St* có độ dài ngắn nhất và xuất hiện đủ 26 kí tự 'a' .. 'z'.

Dữ liệu cho trong file S26.INP gồm một xâu duy nhất *St* (độ dài của *St* không quá 10^5).

Kết quả ghi ra file S26.OUT là độ dài của xâu con ngắn nhất tìm được. Nếu không có xâu con nào chứa đủ các kí tự từ 'a' .. 'z' thì ghi 0.

Ví dụ:

S26.INP	S26.OUT
oabcupdefghijklmnopqrstuivbwaxvyrz	33

**5☀. Độ sắc nét**

Con đường Light Road được trang trí gồm N bóng đèn màu. Các đèn màu được đánh số thứ tự từ $1, 2, \dots, N$ theo dọc tuyến đường. Bóng đèn thứ i có màu được biểu diễn bởi một kí tự latin thường 'a' .. 'z'. Như vậy, dãy N bóng đèn có nhiều nhất 26 màu khác nhau. Độ sắc nét của dãy các bóng đèn liên tiếp từ vị trí L đến vị trí R được định nghĩa:

$$S(L, R) = \frac{C(L, R)}{R-L+1} \text{ trong đó } C(L, R) \text{ là số màu khác nhau của các bóng đèn từ vị trí } L \text{ đến vị trí } R.$$

Ví dụ: $N = 5$, màu các bóng đèn: "aaabc", ta có $S(1, 2) = \frac{1}{2}$; $S(1, 5) = \frac{3}{5}$;

Yêu cầu: Tìm chỉ số L và R sao cho $S(L, R)$ nhỏ nhất.

Dữ liệu cho trong file LightRoad.Inp gồm:

- Dòng đầu ghi số nguyên dương N là số bóng đèn ($N \leq 10^5$).
- Dòng thứ 2 ghi chuỗi kí tự gồm N kí tự biểu diễn màu của N bóng đèn.

Kết quả ghi ra file LightRoad.Out gồm hai chỉ số L và R tìm được (ghi trên một dòng). Nếu có nhiều cặp chỉ số L, R thì đưa ra một cặp chỉ số bất kì.

Ví dụ:

LightRoad.Inp	LightRoad.Out
6 ananas	1 5

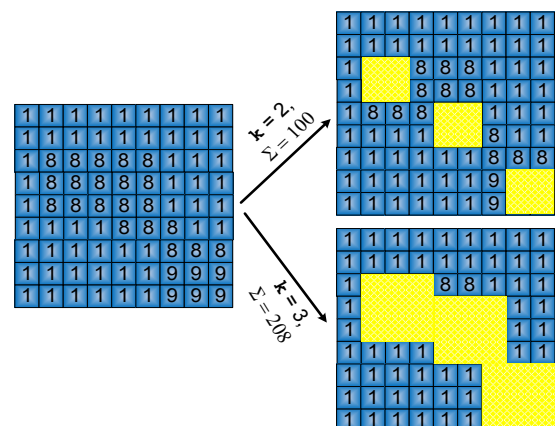
**6☀. Thăm dò dầu mỏ**

Chính phủ Siruseri quyết định tổ chức đấu thầu thăm dò khai thác dầu mỏ ở một vùng thềm lục địa giàu tiềm năng. Vùng biển này có dạng hình chữ nhật, được chia thành $m \times n$ ô vuông (m hàng, n cột, $1 < m, n \leq 1500$).

Tổng cục dầu khí của Siruseri đã thăm dò sơ bộ và đánh giá được trữ lượng tiềm năng ở mỗi ô. Số liệu này được công bố dưới dạng số nguyên không âm, ghi ở mỗi ô và có giá trị không quá 500.

Để tránh nguy cơ độc quyền, chính phủ quy định mỗi công ty tham gia đấu thầu sẽ đăng ký xin khai thác ở một lô hình vuông kích thước $k \times k$ ô. Tập đoàn dầu khí AOE có 3 công ty tham dự đấu thầu. Họ thỏa thuận ngầm với nhau chọn 3 lô riêng biệt sao cho tổng trữ lượng tiềm năng của các lô này là lớn nhất. Trong ví dụ nêu ở hình bên, nếu $k = 2$ thì tập đoàn AOE sẽ nhận được vùng khai thác với tổng tiềm năng là 100, còn với $k = 3 \rightarrow$ tổng tiềm năng sẽ là 208.

Yêu cầu: Cho m, n, k và trữ lượng tiềm năng ở mỗi ô. Hãy xác định tổng tiềm năng AOE có thể giành được. Dữ liệu đảm bảo tồn tại 3 lô riêng biệt.





Dữ liệu: Vào từ file văn bản OIL.INP:

- Dòng đầu tiên chứa 3 số nguyên m , n và k ;
- Dòng thứ i trong m dòng sau chứa n số nguyên không âm mô tả trữ lượng các ô thuộc dòng i .

Kết quả: Đưa ra file văn bản OIL.OUT một số nguyên – tổng tiềm năng AOE có thể giành được.

Ví dụ:

OIL.INP	OIL.OUT
9 9 3	208
1 1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1 1	
1 8 8 8 8 8 1 1 1	
1 8 8 8 8 8 1 1 1	
1 8 8 8 8 8 1 1 1	
1 1 1 1 8 8 8 1 1	
1 1 1 1 1 1 8 8 8	
1 1 1 1 1 1 9 9 9	
1 1 1 1 1 1 9 9 9	