

### Câu 6. Tên bài: GONME.CPP

Ta lưu  $m$  dãy vào ma trận  $A$ ,  $A[i, j]$  lưu số thứ  $j$  trong dãy hoán vị thứ  $i$ . Gọi  $D[i, j]$  là vị trí của số  $j$  trong dãy hoán vị thứ  $i$ ,  $F[i]$  là độ dài dãy con chung dài nhất kết thúc bằng số  $i$ . Với mỗi số thứ  $i$  trong dãy hoán vị đầu tiên, lần lượt ta xét những số thứ  $j$  đứng sau nó, kiểm tra xem cặp số này có thỏa mãn tính thứ tự trong  $m - 1$  dãy còn lại hay không, tức là số  $A[1, i]$  có đứng trước số  $A[1, j]$  trong  $m - 1$  dãy còn lại hay không. Nếu đúng hết thì  $F[A[1, i]] = \text{Max}(F[A[1, i]], F[A[1, j]] + 1)$ . Sau mỗi lần tính chọn lại giá trị lớn nhất cho biến  $res$  và cuối cùng ghi ra  $res$ .

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

### Câu 7: Tên bài: SUBSTRK.CPP

+ Thêm dấu cách vào đầu xâu:  $s = " " + s$ ; để ta xét xâu từ vị trí 1.

+ Gọi  $F[i]$  là số lượng kí tự 1 khi xét xâu  $s$  từ vị trí 1 đến vị trí  $i$ , ta có công thức tính

$$+ F[i] = F[i-1] + 1; \text{ nếu } s[i] = '1'$$

$$+ \text{ Ngược lại } F[i] = F[i-1];$$

Như vậy, ta có  $F[j] - F[i]$  chính là số lượng kí tự 1 của xâu  $s$  xét từ vị trí  $i+1$  đến vị trí  $j$ . Giả sử  $n$  là vị trí cuối của xâu. Ta duyệt  $i$  từ 0 đến  $n$ , với mỗi vị trí của  $i$ : ta tìm  $left$  là vị trí phía sau  $i$  mà là vị trí đầu tiên thỏa mãn  $f[left] - f[i] = k$ ; Nếu không có  $left$  thỏa mãn thì ta quay lên tiếp tục lần lặp tiếp theo. Khi tìm được  $left$  thì ta tìm tiếp  $right$  là vị trí cuối cùng thỏa mãn  $f[left] - f[i] = k$ ; Từ đây là cập nhật thêm vào kết quả:  $right - left + 1$ ; (vì trong các đoạn này sẽ có đúng  $k$  kí tự 1)

Độ phức tạp của thuật toán là  $O(n \log n)$

### Câu 8. Tên bài: BLAST.CPP

+ Gọi  $n$  là độ dài xâu  $A$ ,  $m$  là độ dài xâu  $B$

+ Gọi  $F[i, j]$  là khoảng cách nhỏ nhất giữa hai xâu mở rộng mà khi xét xâu  $A$  đến vị trí thứ  $i$  và xâu  $B$  đến vị trí thứ  $j$ .

$$+ \text{ Nhận thấy } F[0, 0] = 0; F[i, 0] = i * k; F[0, j] = j * k;$$

+ Công thức quy hoạch động

$$F[i, j] := \min(F[i, j-1] + k, F[i-1, j] + k, F[i-1, j-1] + 2 * k, F[i-1, j-1] + |a[i] - b[j]|)$$

+ Kết quả của bài toán là  $F[n, m]$

**Câu 9. Tên bài: STMERGE.CPP**

+ Gọi  $F[i, j, 0]$  là tổng nhỏ nhất tạo thành từ xâu  $X[1..i]$  và  $Y[1..j]$ , trong đó kí tự cuối là  $X[j]$ .  $F[i, j, 1]$  là tổng nhỏ nhất tạo thành từ xâu  $X[1..i]$  và  $Y[1..j]$ , trong đó kí tự cuối là  $Y[j]$ .

+ Nhận thấy  $f[i, 0, 0] = 0$ ;  $f[0, j, 1] = 0$ ;

+ Công thức:  $f[i, j, 0] := \min(f[i-1, j, 0], f[i-1, j, 1] + c[i, j])$ ;

$f[i, j, 1] := \min(f[i, j-1, 1], f[i, j-1, 0] + c[i, j])$ ;

+ Kết quả bài toán  $\min(f[m, n, 0], f[m, n, 1])$ . Độ phức tạp thuật toán là  $O(m*n)$

**Câu 10. Tên bài: SUBSTR**

- Thay vì ta đi tìm xóa số kí tự ít nhất để xâu trở thành xâu đặc biệt, thì ta đi tìm xâu con có độ dài lớn nhất thỏa mãn yêu cầu đề bài:

- $n \leq 20$  ta có thể làm bằng cách đệ quy sinh nhị phân để xét tất cả các xâu con, với mỗi xâu ta đi kiểm tra nó có thỏa mãn đề bài hay không, và cập nhật độ dài xâu vào kết quả. Lấy  $n$  trừ đi độ dài đó là ra kết quả bài toán.

- Với  $n$  lớn hơn ta làm tương tự như bài xâu con chung:

+ Gọi  $f(i, j)$  là độ dài xâu con lớn nhất thỏa mãn đề bài mà 2 đầu là nguyên âm khi xét xâu từ vị trí  $i$  đến vị trí  $j$ .

+ Gọi  $g(i, j)$  là độ dài xâu con lớn nhất thỏa mãn đề bài mà 2 đầu là phụ âm khi xét xâu từ vị trí  $i$  đến vị trí  $j$ .

- $f(i, j) = \max(f(i, j-1), f(i+1, j), g(i+1, j-1) + 2)$  nếu 2 đầu  $i, j$  là nguyên âm

- $g(i, j) = \max(g(i, j-1), g(i+1, j), f(i+1, j-1) + 2)$  nếu 2 đầu  $i, j$  là phụ âm

Kết quả bài toán là:  $n - \max(f(1, n), g(1, n))$ . Độ phức tạp bài toán là  $O(n^2)$