**Student name: Ho Ping Chong**

**Student number: 1155057016**

**Date: 3 Oct 2014**

**Title: GNBF5010 assignment 1**

1. Several ways to write Fibonacci series in C language
   * 1. For loop

#include<stdio.h>

int main()

{

int n, first = 0, second = 1, next, c;

printf("Enter the number of terms\n");

scanf("%d",&n);

printf("First %d terms of Fibonacci series are :-\n",n);

for ( c = 0 ; c < n ; c++ )

{

if ( c <= 1 )

next = c;

else

{

next = first + second;

first = second;

second = next;

}

printf("%d\n",next);

}

return 0;

}

* + 1. Recursive function

#include<stdio.h>

int Fibonacci(int);

main()

{

int n, i = 0, c;

printf("Enter the number of terms\n");

scanf("%d",&n);

printf("First %d terms of Fibonacci series are :-\n",n);

for ( c = 1 ; c <= n ; c++ )

{

printf("%d\n", Fibonacci(i));

i++;

}

return 0;

}

int Fibonacci(int n)

{

if ( n == 0 )

return 0;

else if ( n == 1 )

return 1;

else

return ( Fibonacci(n-1) + Fibonacci(n-2) );

}

* + 1. Dynamic programming

int fib(int n)

{

/\* Declare an array to store fibonacci numbers. \*/

int f[n+1];

int i;

/\* 0th and 1st number of the series are 0 and 1\*/

f[0] = 0;

f[1] = 1;

for (i = 2; i <= n; i++)

{

/\* Add the previous 2 numbers in the series

and store it \*/

f[i] = f[i-1] + f[i-2];

}

return f[n];

}

int main ()

{int n,c,j=0;

printf("Enter the number of terms\n");

scanf("%d",&n);

printf("First %d terms of Fibonacci series are :-\n",n);

for ( c = 1 ; c <= n ; c++ )

{

printf("%d\n", fib(j));

j++;

}

getchar();

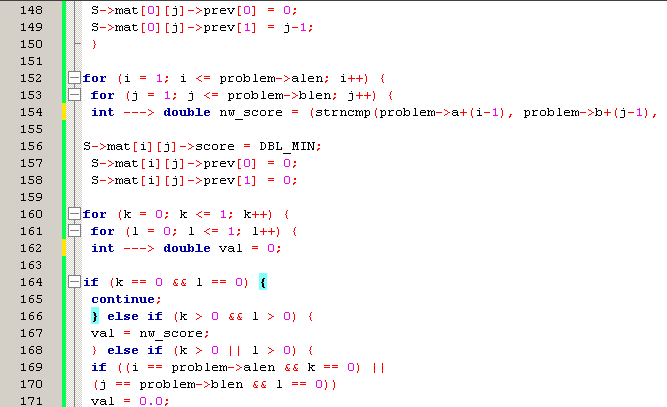
return 0;

}

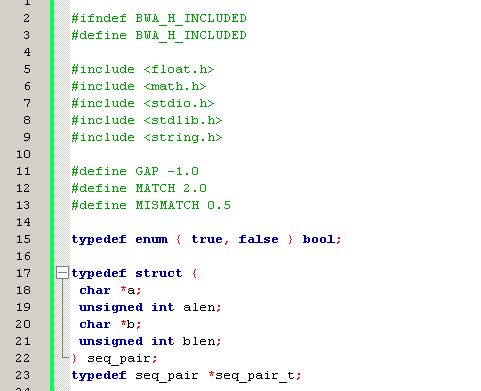
Performance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methodology | 10 nos | 40 nos | 160 nos | 640 nos |
| For loop | 2.512s | 2.761s | 2.824s | 3.079s |
| Recursive function | 2.695s | 3.954s | >60s | >60s |
| Dynamic programming | 2.476s | 2.749s | 2.960s | 4.720s |

In file swa.c line 154 & 162 amend data type from integer to double thus to adopt mismatch score with 0.5.

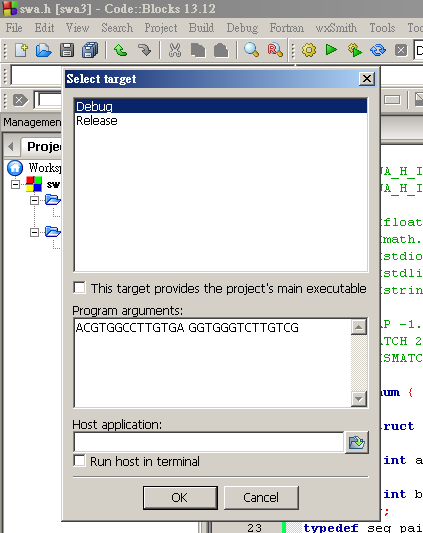


In swa.h file line 11, 12 and 13 amend the score as in the question as -1.0, 2.0 and 0.5.

U

Set argument in Code::blocks by select project>Set project’s argument.

Input the sequences.

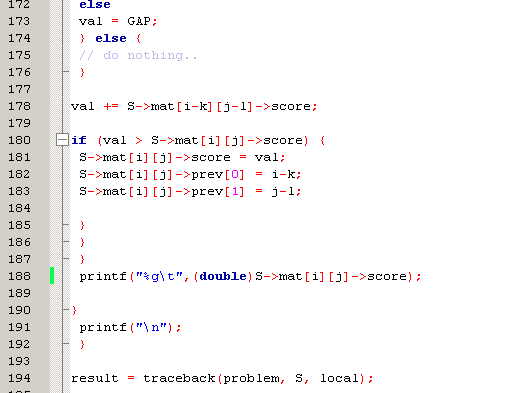


Go to build and run

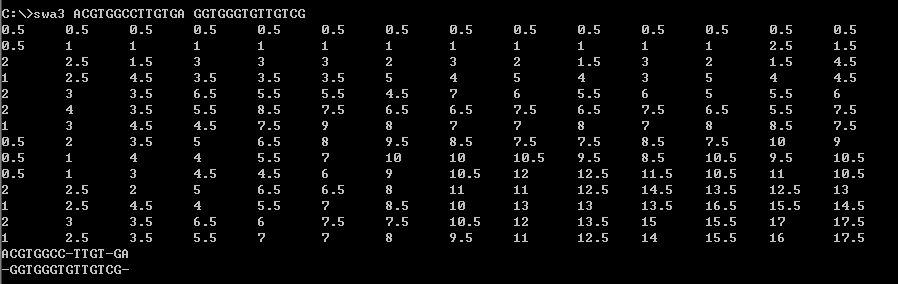
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | C | G | T | G | G | C | - | C | T | T | G | T | - | G | A |
| - | G | G | T | G | G | G | T | C | T | T | G | T | C | G | - |

Process returned 0 (0x0) execution time : 0.033 s

To provide decimal display in the matrix, change line 188 “%d” to “%g” and “int” to “double” so as to store and display decimal instead of integer .



In command console, input “mode 120,50” to expand column size and run swa.exe with argument “ACGTGGCCTTGTGA” and “GGTGGGTGTTGTCG”.



Source code as following: amended code is in red

swa.c

#include "swa3.h"

/\* reverse a string in place, return str \*/

static char\* reverse(char \*str) {

char \*left = str;

char \*right = left + strlen(str) - 1;

char tmp;

while (left < right) {

tmp = \*left;

\*(left++) = \*right;

\*(right--) = tmp;

}

return str;

}

// works globally

static seq\_pair\_t traceback(seq\_pair\_t problem, matrix\_t S, bool local) {

seq\_pair\_t result = malloc(sizeof(seq\_pair));

unsigned int i = S->m - 1;

unsigned int j = S->n - 1;

unsigned int k = 0;

char c[S->m + S->n + 1];

char d[S->m + S->n + 1];

memset(c, '\0', sizeof(c));

memset(d, '\0', sizeof(d));

if (local == true) {

unsigned int l, m;

double max = FLT\_MIN;

for (l = 0; l < S->m; l++) {

for (m = 0; m < S->m; m++) {

if (S->mat[l][m]->score > max) {

i = l;

j = m;

}

}

}

}

if (S->mat[i][j]->prev[0] != 0 && S->mat[i][j]->prev[1] != 0) {

while (i > 0 || j > 0) {

unsigned int new\_i = S->mat[i][j]->prev[0];

unsigned int new\_j = S->mat[i][j]->prev[1];

if (new\_i < i)

\*(c+k) = \*(problem->a+i-1);

else

\*(c+k) = '-';

if (new\_j < j)

\*(d+k) = \*(problem->b+j-1);

else

\*(d+k) = '-';

k++;

i = new\_i;

j = new\_j;

}

}

result->a = malloc(sizeof(char) \* k + 1);

result->b = malloc(sizeof(char) \* k + 1);

memset(result->a, '\0', sizeof(\*result->a));

memset(result->b, '\0', sizeof(\*result->b));

reverse(c);

reverse(d);

strcpy(result->a, c);

strcpy(result->b, d);

result->alen = k;

result->blen = k;

return result;

}

static matrix\_t create\_matrix(unsigned int m, unsigned int n) {

matrix\_t S = malloc(sizeof(matrix));

unsigned int i, j;

S->m = m;

S->n = n;

S->mat = malloc(sizeof(entry\_t) \* m \* n);

for (i = 0; i < m; i++) {

S->mat[i] = malloc(sizeof(entry\_t) \* n);

}

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

S->mat[i][j] = malloc(sizeof(entry));

}

}

return S;

}

void destroy\_matrix(matrix\_t S) {

unsigned int i, j;

for (i = 0; i < S->m; i++) {

for (j = 0; j < S->n; j++) {

free(S->mat[i][j]);

}

}

free(S);

return;

}

void destroy\_seq\_pair(seq\_pair\_t pair) {

free(pair->a);

free(pair->b);

free(pair);

return;

}

static seq\_pair\_t smith\_waterman(seq\_pair\_t problem, bool local) {

unsigned int m = problem->alen + 1;

unsigned int n = problem->blen + 1;

matrix\_t S = create\_matrix(m, n);

seq\_pair\_t result;

unsigned int i, j, k, l;

S->mat[0][0]->score = 0;

S->mat[0][0]->prev[0] = 0;

S->mat[0][0]->prev[1] = 0;

for (i = 1; i <= problem->alen; i++) {

S->mat[i][0]->score = 0.0;

S->mat[i][0]->prev[0] = i-1;

S->mat[i][0]->prev[1] = 0;

}

for (j = 1; j <= problem->blen; j++) {

S->mat[0][j]->score = 0.0;

S->mat[0][j]->prev[0] = 0;

S->mat[0][j]->prev[1] = j-1;

}

for (i = 1; i <= problem->alen; i++) {

for (j = 1; j <= problem->blen; j++) {

double nw\_score = (strncmp(problem->a+(i-1), problem->b+(j-1), 1) == 0) ? MATCH : MISMATCH;

S->mat[i][j]->score = DBL\_MIN;

S->mat[i][j]->prev[0] = 0;

S->mat[i][j]->prev[1] = 0;

for (k = 0; k <= 1; k++) {

for (l = 0; l <= 1; l++) {

double val = 0;

if (k == 0 && l == 0) {

continue;

} else if (k > 0 && l > 0) {

val = nw\_score;

} else if (k > 0 || l > 0) {

if ((i == problem->alen && k == 0) ||

(j == problem->blen && l == 0))

val = 0.0;

else

val = GAP;

} else {

// do nothing..

}

val += S->mat[i-k][j-l]->score;

if (val > S->mat[i][j]->score) {

S->mat[i][j]->score = val;

S->mat[i][j]->prev[0] = i-k;

S->mat[i][j]->prev[1] = j-l;

}

}

}

printf("%g\t",(double)S->mat[i][j]->score);

}

printf("\n");

}

result = traceback(problem, S, local);

destroy\_matrix(S);

return result;

}

int main(int argc, const char \*\*argv) {

if (argc != 3) {

printf("Wrong Input Format!!!!!!!\nEnter ./a.out SEQ1 SEQ2\n");

exit(1);

}

{

seq\_pair problem;

seq\_pair\_t result;

char c[strlen(argv[1])], d[strlen(argv[2])];

strcpy(c, argv[1]);

strcpy(d, argv[2]);

problem.a = c;

problem.alen = strlen(problem.a);

problem.b = d;

problem.blen = strlen(problem.b);

result = smith\_waterman(&problem, false);

printf("%s\n%s\n", result->a, result->b);

}

exit(0);

}

swa.h

#ifndef BWA\_H\_INCLUDED

#define BWA\_H\_INCLUDED

#include <float.h>

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define GAP -1.0

#define MATCH 2.0

#define MISMATCH 0.5

typedef enum { true, false } bool;

typedef struct {

char \*a;

unsigned int alen;

char \*b;

unsigned int blen;

} seq\_pair;

typedef seq\_pair \*seq\_pair\_t;

typedef struct {

double score;

unsigned int prev[2];

} entry;

typedef entry \*entry\_t;

typedef struct {

unsigned int m;

unsigned int n;

entry\_t \*\*mat;

} matrix;

typedef matrix \*matrix\_t;

static char\* reverse(char \*str);

static seq\_pair\_t traceback(seq\_pair\_t problem, matrix\_t S, bool local);

static matrix\_t create\_matrix(unsigned int m, unsigned int n);

void destroy\_matrix(matrix\_t S);

void destroy\_seq\_pair(seq\_pair\_t pair);

static seq\_pair\_t smith\_waterman(seq\_pair\_t problem, bool local);

#endif // BWA\_H\_INCLUDED

I am submitting the assignment for:

□ an individual project or

□ a group project on behalf of all members of the group. It is hereby confirmed that the submission is authorized by all members of the group, and all members of the group are required to sign this declaration.

I/We declare that the assignment here submitted is original except for source material explicitly acknowledged, the piece of work, or a part of the piece of work has not been submitted for more than one purpose (i.e. to satisfy the requirements in two different courses) without declaration, and that the submitted soft copy with details listed in the <Submission Details> is identical to the hard copy(ies), if any, which has(have) been / is(are) going to be submitted. I/We also acknowledge that I am/we are aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the University website [http://www.cuhk.edu.hk/policy/academichonesty/](http://www.cuhk.edu.hk/policy/academichonesty). In the case of a group project, we are aware that each student is responsible and liable to disciplinary actions should there be any plagiarized contents in the group project, irrespective of whether he/she has signed the declaration and whether he/she has contributed directly or indirectly to the plagiarized contents.

It is also understood that assignments without a properly signed declaration by the student concerned and in the case of a group project, by all members of the group concerned, will not be graded by the teacher(s).

\_\_\_\_Ho Ping Chong\_\_ \_\_\_\_3 Oct 2014\_\_\_\_\_\_

Signature(s) Date

Ho Ping Chong\_\_\_ \_\_\_1155057016\_\_\_\_\_

Name(s) Student ID(s)

\_\_\_\_GNBF5010\_\_\_\_\_\_\_ \_\_Introduction to programming\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course code Course title