

Assignment 3 of Algorithm Design and Analysis

L.J.SHOU 寿林钧
No. 201228013229133

Nov.07 2012

1 Dynamic Programming(6)

My implementation results are shown as follows using two different scoring functions.

Implementaion details in the Appendix.

$$\delta_1(a, b) = \begin{cases} 1; & a = b \\ -1; & a \neq b \\ -3; & a = '-' \text{ or } b = '-' \end{cases} \quad (1)$$

The optimal score we obtain is 618.

S'=

```
CGCTTCCAGCAGTGACGGAAGGCCTCTTCGGAGTCTCGAGTGGCGAACGGGTGA
GTAACACGTGGGCAATCTGCCCTGCACTTCGGGATAAGCCTGGGAAACTGGGTC
TAATACCGGATAGGACCTCAAGACGCATGTCTTCTGGTGAAAGCTTTTGCGGT
GTGGGATGGGCCCCGCGCCTATCAGCTTGTTGGTGGGGTGACGGCCTACCAAGG
CGACGACGGGTAGCCGGCCTGAGAGGGTGTCGGGCCACACTGGGACTGAGATAC
GGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGCAAAG
CCTGATGCAGCGACGCCGCGTGGGGGATGACGGCCTTCGGGTTGTAAACCTCTT
TCACCATCGACGAAGGTCCGGGTTTCTCGGATTGACGGTAGGTGGAGAAGAAG
CACCGGCCAACTACGTGCCAGCAGCCGCGGTAATACGTAGGGTGCGAGCGTTGT
CCGGAATTA-CTGGGCGTAAAGAGCTCGTAGGTGTTTTGTGCGGTTGTTTCGTG
AAATCTCACGGCTTAACGTGAGCGTGCGGGCGATAC-GGGCAGACTAGAGTA
CTGCAGGGGAGACTGGAATTCCTGGTGAGCGGTGGAATGCGCAGATATCAGGA
GGAACACCGGT-GGCGAAGGCGGGTCTCTGGGCAGTAACTGACGCTGAGGAGC
GAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGTAG-TCCACGCCGTAAaA
```

CGGTGGGTACTAGGTGTGGGTTTCCTTCCTTGGGATCCGTGCCGTAGCTAACGC
ATTAAGTACCCCG-CCTGGGGAGTACGGCCGCAAGGCTAAA-CTCATGTAA

$\mathbf{T}' =$

-G-TTCCAGCAGTGACGGAAGGCC-CTTCGGGGTCTCGAGTGGCGAACGGGTGA
GTAACACGTGGGTGATCTGCCCTGCACTTTGGGATAAGCCTGGGAACTGGGTC
TAATACCGAATATGACCACGCGCTTCATGGTGTGTGGTGAAAAGCTTTTGCGGT
GTGGGATGGGCCCCGCGCCTATCAGCTTGTTGGTGGGGTAATGGCCTACCAAGG
CGACGACGGGTAGCCGGCCTGAGAGGGTGACCGGCCACACTGGGACTGAGATAC
GGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGCAAG
CCTGATGCAGCGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTT
TCAATAGGGACGAAGCGCAAG-TGA-CGG-T-ACC-TA-T-AGAAGAAG
GACCGGCCAACTACGTGCCAGCAGCCGCGTAATACGTAGGGTCCGAG-CGTT
GTCCGAATTACTGGGCGTAAAGAGCTCGTAGGTGGTTTGTTCGCGTTGTTTCGTG
AAAACCTCACAGCTTAACTGTGG-GCGTGCGGGCGATACGGGCAGACTAGAGTA
CTGCAGGGGAGACTGGAATTCCTGGTGTAGCGGTGGAATGCGCAGATATC-AG
GAGGAACACCGGTGGCGAAGGCGGGTCTCTGGGCAGTAACTGACGCTGAGGAGC
GAAAGCGTGGGGAGCGAACAGGAT-TAGATACCCTGGTAGTCCACGCCGTAAA
CGGTGGGTACTAGGTGTGGGTTTCCTTCCTTGGGATCCGTGCCGTAGCTAAC-
GCATTAAGTACCCCGCCTGGGGAGTACGGCCGCAAGGCTAAAACCTCAAATA-

$$\delta_2(a, b) = \begin{cases} 2; & a = b \\ -1; & a \neq b \\ -2; & a = '-' \text{ or } b = '-' \end{cases} \quad (2)$$

The optimal score is 1449.

$\mathbf{s}' =$

CGCTTCCAGCAGTGACGGAAGGCCCTTCGAGTCTCGAGTGGCGAACGGGT
GAGTAACACGTGGGCAATCTGCCCTGCACTTC-GGGATAAGCCTGGGAAAC
TGGGTCTAATACCGGATAGGACCTCAAGACG-CATG-TCTTCTGGTGAAAAG
CTTTTGCGGTG-TGGGATGGGCCCCGCGGCCTATCAGCTTGTTGGTGGGGTG
ACGGCCTACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGG-TGTCCGGCC
ACACTGGGACTGAGATACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAA
TATTGCACAATGGGCGCAA-GCCTGATGCAGCGACGCCGCGTGGGGATGA
CGGCCTTCGGGTTGTAAACCTCTTTCACCATCGACGAAG-GTCCGGGTTTTC
TCGGATTGACGGTAGGTGGAGAAGAAGCACCGGCCAACTACGTGCCAGCAGC
CGCGGTAATACGTAGGGTGCGAGCGTTGTCCGGAATTA-CTGGGCGTAAAG
AGCTCGTAGGTGGTTTGTTCGCGTTGTTTCGTGAAATCTCACGGCTTAACTGTG
AGCGTGCGGGCGATAC-GGGCAGACTAGAGTACTGCAGGGGAGACTGGAAT
TCCTGGTGTAGCGGTGGAATGCGCAGATATCAGGAGGAACACCGGT-GGCG

AAGGCGGGTCTCTGGGCAGTAAC TGACGCTGAGGAGCGAAAGCGTGGGGAGC
 GAACAGGATTAGATACCCTGGTAG-TCCACGCCGTAAACGGTGGGTACTAG
 GTGTGGGTTTCCTTCCTTGGGATCCGTGCCGTAGCTAACGCATTAAGTACCC
 CG-CCTGGGGAGTACGGCCGCAAGGCTAAA-CTCATGTAA

T'=

-G-TTCCAGCAGTGACGGAAGGCC-CTTCGGGGTCTCGAGTGGCGAACGGGT
 GAGTAACACGTGGGTGATCTGCCCTGCA-CTTTGGGATAAGCCTGGGAAAC
 TGGGTCTAATACCGAATATGACCACGCG-CTTCATGGTGTG-TGGTGGAAAG
 CTTTTCGCG-GTGTGGGATGGGCCCCGGCCCTATCAGCTTGTGTTGGTGGGGTA
 ATGGCCTACCAAGCGACGACGGGTAGCCGGCCTGAGA-GGGTGACCGGCC
 ACACTGGGACTGAGATACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAA
 TATTGCACAATGGGCG-CAAGCCTGATGCAGCGACGCCGCGTGAGGGATGA
 CGGCCCTTCGGGTTGTAAACCTCTTTCAATAGGGACGAAGCG-CAAG-TGA
 -CGG-T-ACC-TA-T-AGAAGAAGGACCGGCCAACTACGTGCCAGCAGC
 CGCGGTAATACGTAGGGTCCGAG-CGTTGTCCGGAATTACTGGGCGTAAAG
 AGCTCGTAGGTGGTTTGTGCGTTGTTCGTGAAAACCTCACAGCTTAAGTGTG
 G-GCGTGCGGGCGATACGGGCAGACTAGAGTACTGCAGGGGAGACTGGAAT
 TCCTGGTGTAGCGGTGGAATGCGCAGATATC-AGGAGGAACACCGGTGGCG
 AAGGCGGGTCTCTGGGCAGTAAC TGACGCTGAGGAGCGAAAGCGTGGGGAGC
 GAACAGGAT-TAGATACCCTGGTAGTCCACGCCGTAAACGGTGGGTACTAG
 GTGTGGGTTTCCTTCCTTGGGATCCGTGCCGTAGCTAAC-GCATTAAGTAC
 CCCGCTGGGGAGTACGGCCGCAAGGCTAAAAC TCAAATA-

From these two instances, we can see that the optimal alignment and score highly depend on how you design the scoring scheme.

2 Appendix

```

1  /*implementation of Needleman-Wunch Algorithm in C by L.J.SHOU */
2  #include<stdio.h>
3  #include<stdlib.h>
4  #include<malloc.h>
5  #include<string.h>
6  #define score1 2
7  #define score2 -1
8  #define score3 -2
9  void get_opt_alignment1(int **OPT,int m,int n,char *s) //backtracking
10 {
11     if(m<1&& n<1)
12         return;
13     if(m<1) //in the first row
14     {
15         get_opt_alignment1(OPT,m,n-1,s);
16         printf("—");
17     }
18     else if(n<1) // in the first column
19     {
20         get_opt_alignment1(OPT,m-1,n,s);
21         printf("%c",s[m-1]);
22     }
23     else
24     {
25         if(OPT[m][n]==OPT[m][n-1]+score3)
26         {
27             get_opt_alignment1(OPT,m,n-1,s);
28             printf("—");
29         }
30         else if(OPT[m][n]==OPT[m-1][n]+score3)
31         {
32             get_opt_alignment1(OPT,m-1,n,s);
33             printf("%c",s[m-1]);
34         }
35         else
36         {
37             get_opt_alignment1(OPT,m-1,n-1,s);
38             printf("%c",s[m-1]);
39         }
40     }
41 }
42 void get_opt_alignment2(int **OPT,int m,int n,char *t)
43 {
44     if(m<1&& n<1)
45         return;
46     if(m<1)
47     {
48         get_opt_alignment2(OPT,m,n-1,t);
49         printf("%c",t[n-1]);
50     }
51     else if(n<1)
52     {
53         get_opt_alignment2(OPT,m-1,n,t);
54         printf("—");
55     }
56     else
57     {
58         if(OPT[m][n]==OPT[m-1][n]+score3)
59         {
60             get_opt_alignment2(OPT,m-1,n,t);
61             printf("—");
62         }
63         else if(OPT[m][n]==OPT[m][n-1]+score3)
64         {
65             get_opt_alignment2(OPT,m,n-1,t);
66             printf("%c",t[n-1]);
67         }
68         else
69         {
70             get_opt_alignment2(OPT,m-1,n-1,t);
71             printf("%c",t[n-1]);
72         }
73     }

```

```

74 }
75 int Needleman_Wunch(char s[], char t[])
76 {
77     int i, j, m, n, **OPT, opt;
78     m=strlen(s);
79     n=strlen(t);
80     OPT=(int **) malloc((m+1)*sizeof(int *));
81     for(i=0; i<m+1; i++)
82         OPT[i]=(int *) malloc((n+1)*sizeof(int));
83     for(i=0; i<m+1; i++)
84         OPT[i][0]=score3*i;
85     for(j=0; j<n+1; j++)
86         OPT[0][j]=score3*j;
87     for(i=1; i<m+1; i++)
88         for(j=1; j<n+1; j++)
89         {
90
91             if(s[i-1]==t[j-1])
92                 OPT[i][j]=OPT[i-1][j-1]+score1;
93             else
94                 OPT[i][j]=OPT[i-1][j-1]+score2;
95             if(OPT[i-1][j]+score3>OPT[i][j])
96                 OPT[i][j]=OPT[i-1][j]+score3;
97             if(OPT[i][j-1]+score3>OPT[i][j])
98                 OPT[i][j]=OPT[i][j-1]+score3;
99         }
100     printf("S'=\n");
101     get_opt_alignment1(OPT, m, n, s);
102     printf("nT'=\n");
103     get_opt_alignment2(OPT, m, n, t);
104     opt=OPT[m][n];
105     //free the assigned memory
106     for(i=0; i<m+1; i++)
107         free(OPT[i]);
108     free(OPT);
109     return opt;
110 }
111 void main()
112 {
113     FILE *fp;
114     char *s, *t;
115     char s1[]="urr", t1[]="ocurrence";
116     //=====Load Gene1_Seq.txt and Gene2_Seq.txt
117     int m=0, n=0, i;
118     //m and n denotes the lenhth of Gene1_seq and Gene2_seq respectively
119     if((fp=fopen("Gene1_Seq.txt", "r"))==NULL)
120     {
121         printf("not open");
122         exit(0);
123     }
124     while (fgetc(fp)!=EOF)
125         m++;
126     s=(char *) malloc(sizeof(char)*(m+1)); //the last element is '\0'
127     rewind(fp);
128     i=0;
129     while((s[i++]=fgetc(fp))!=EOF);
130     s[m]='\0';
131     if((fp=fopen("Gene2_Seq.txt", "r"))==NULL)
132     {
133         printf("not open");
134         exit(0);
135     }
136     while (fgetc(fp)!=EOF)
137         n++;
138     t=(char *) malloc(sizeof(char)*(n+1));
139     rewind(fp);
140     i=0;
141     while((t[i++]=fgetc(fp))!=EOF);
142     t[n]='\0';
143     fclose(fp);
144     printf("n\nThe optimal score is %d.\n", Needleman_Wunch(s, t));
145 }

```