Dynamic Programming: Longest Common Subsequence

Class 29

Protein

- a protein is a complex molecule composed of long single-strand chains of amino acid molecules
- there are 20 amino acids that make up proteins
- the primary structure of a protein is the sequence of its amino acids

Protein

- a protein is a complex molecule composed of long single-strand chains of amino acid molecules
- there are 20 amino acids that make up proteins
- the primary structure of a protein is the sequence of its amino acids

- given a new unknown protein, it is relatively easy to determine the primary structure
- since there are 20 amino acids, and 26 Roman letters, we represent a protein's primary structure as a string of letters

Protein

- a protein is a complex molecule composed of long single-strand chains of amino acid molecules
- there are 20 amino acids that make up proteins
- the primary structure of a protein is the sequence of its amino acids

- given a new unknown protein, it is relatively easy to determine the primary structure
- since there are 20 amino acids, and 26 Roman letters, we represent a protein's primary structure as a string of letters

A alanine
C cysteine
D aspartic acid
E glutamic acid
etc.

A Protein

- human breast cancer susceptibility protein 1:

MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTKCDHIFCKFCMLKLLNQKKGPSQCPLCKNDITKRSLQESTRFSQLVEELLKIICAFQ
LDTGLEYANSYNFAKKENNSPEHLKDEVSIIQSMGYRNRAKRLLQSEPENPSLQETSLSVQLSNLGTVRTLRTKQRIQPQKTSVYIELGSDSSE
DTVNKATYCSVGDQELLQITPQGTRDEISLDSAKKAACEFSETDVTNTEHHQPSNNDLNTTEKRAAERHPEKYQGSSVSNLHVEPCGTNTHASS
LQHENSSLLLTKDRMNVEKAEFCNKSKQPGLARSQHNRWAGSKETCNDRRTPSTEKKVDLNADPLCERKEWNKQKLPCSENPRDTEDVPWITLN
SSIQKVNEWFSRSDELLGSDDSHDGESESNAKVADVLDVLNEVDEYSGSSEKIDLLASDPHEALICKSERVHSKSVESNIEDKIFGKTYRKKAS
LPNLSHVTENLIIGAFVTEPQIIQERPLTNKLKRKRRPTSGLHPEDFIKKADLAVQKTPEMINQGTNQTEQNGQVMNITNSGHENKTKGDSIQN
EKNPNPIESLEKESAFKTKAEPISSSISNMELELNIHNSKAPKKNRLRRKSSTRHIHALELVVSRNLSPPNCTELQIDSCSSSEEIKKKKYNQM
PVRHSRNLQLMEGKEPATGAKKSNKPNEQTSKRHDSDTFPELKLTNAPGSFTKCSNTSELKEFVNPSLPREEKEEKLETVKVSNNAEDPKDLML
SGERVLQTERSVESSSISLVPGTDYGTQESISLLEVSTLGKAKTEPNKCVSQCAAFENPKGLIHGCSKDNRNDTEGFKYPLGHEVNHSRETSIE
MEESELDAQYLQNTFKVSKRQSFAPFSNPGNAEEECATFSAHSGSLKKQSPKVTFECEQKEENQGKNESNIKPVQTVNITAGFPVVGQKDKPVD
NAKCSIKGGSRFCLSSQFRGNETGLITPNKHGLLQNPYRIPPLFPIKSFVKTKCKKNLLEENFEEHSMSPEREMGNENIPSTVSTISRNNIREN
VFKEASSSNINEVGSSTNEVGSSINEIGSSDENIQAELGRNRGPKLNAMLRLGVLQPEVYKQSLPGSNCKHPEIKKQEYEEVVQTVNTDFSPYL
ISDNLEQPMGSSHASQVCSETPDDLLDDGEIKEDTSFAENDIKESSAVFSKSVQKGELSRSPSPFTHTHLAQGYRRGAKKLESSEENLSSEDEE
LPCFQHLLFGKVNNIPSQSTRHSTVATECLSKNTEENLLSLKNSLNDCSNQVILAKASQEHHLSEETKCSASLFSSQCSELEDLTANTNTQDPF
LIGSSKQMRHQSESQGVGLSDKELVSDDEERGTGLEENNQEEQSMDSNLGEAASGCESETSVSEDCSGLSSQSDILTTQQRDTMQHNLIKLQQE
MAELEAVLEQHGSQPSNSYPSIISDSSALEDLRNPEQSTSEKAVLTSQKSSEYPISQNPEGLSADKFEVSADSSTSKNKEPGVERSSPSKCPSL
DDRWYMHSCSGSLQNRNYPSQEELIKVVDVEEQQLEESGPHDLTETSYLPRQDLEGTPYLESGISLFSDDPESDPSEDRAPESARVGNIPSSTS
ALKVPQLKVAESAQSPAAAHTTDTAGYNAMEESVSREKPELTASTERVNKRMSMVVSGLTPEEFMLVYKFARKHHITLTNLITEETTHVVMKTD
AEFVCERTLKYFLGIAGGKWVVSYFWVTQSIKERKMLNEHDFEVRGDVVNGRNHQGPKRARESQDRKIFRGLEICCYGPFTNMPTDQLEWMVQL
CGASVVKELSSFTLGTGVHPIVVVQPDAWTEDNGFHAIGQMCEAPVVTREWVLDSVALYQCQELDTYLIPQIPHSHY

Similarity

- however, while the primary sequence is necessary for understanding function, it is not sufficient
- an approach to understanding function is to compare the unknown sequence with the sequences of proteins whose functions are known
- two proteins with similar sequences may have similar functions
- the question is the same as with string alignment: what is similar?
- what metric do we use?

Similarity

- however, while the primary sequence is necessary for understanding function, it is not sufficient
- an approach to understanding function is to compare the unknown sequence with the sequences of proteins whose functions are known
- two proteins with similar sequences may have similar functions
- the question is the same as with string alignment: what is similar?
- what metric do we use?

- we can use string alignment
- for proteins, another common similarity metric is measuring the longest common subsequence

Longest Common Subsequence

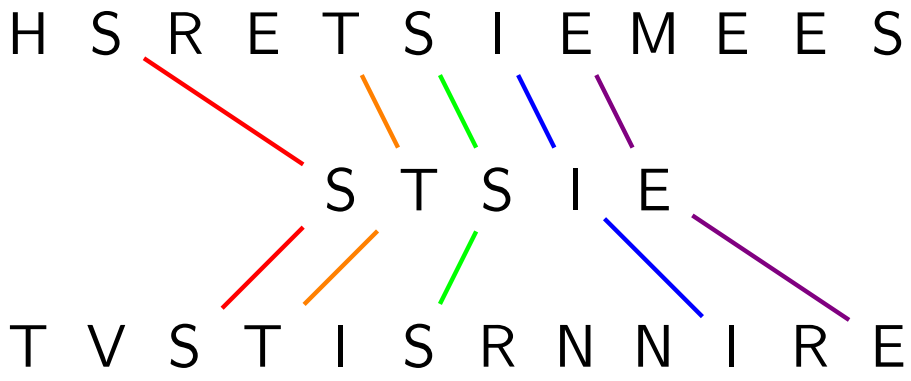
- what is the longest common list of letters, in order, in these sequences?

H S R E T S I E M E E S

T V S T I S R N N I R E

Longest Common Subsequence

- what is the longest common list of letters, in order, in these sequences?



Step 1

- given two sequences

$$s[0..n-1] \text{ and } t[0..m-1]$$

- we wish to find a longest common subsequence, i.e., a subsequence of s:

$$s[i_0], \ldots, s[i_{k-1}]$$

and of t

$$t[j_0], \ldots, t[j_{k-1}]$$

such that

$$s[i_0] = t[j_0], \ldots, s[i_{k-1}] = t[j_{k-1}]$$

where k is as long as possible

Step 1

- given two sequences

$$s[0..n-1] \text{ and } t[0..m-1]$$

- we wish to find a longest common subsequence, i.e., a subsequence of s:

$$s[i_0], \ldots, s[i_{k-1}]$$

and of t

$$t[j_0], \ldots, t[j_{k-1}]$$

such that

$$s[i_0] = t[j_0], \ldots, s[i_{k-1}] = t[j_{k-1}]$$

where k is as long as possible

- let $\text{opt}(i, j)$ be the optimum length of a common subsequence of

$$s[0..i] \text{ and } t[0..j]$$

where $0 \leq i < n$ and $0 \leq j < m$

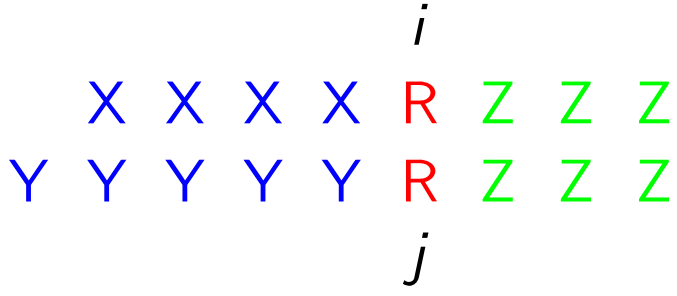
Step 2

- we need a recurrence relation for $\mathrm{opt}(i, j)$
- there are two possibilities:
  1. $s[i] = t[j]$
  2. $s[i] \neq t[j]$

Step 2

- we need a recurrence relation for opt(i, j)
- there are two possibilities:
  1. $s[i] = t[j]$
  2. $s[i] \neq t[j]$

- let's look at case 1
- if the i and j characters of s and t match, then the lcs of the strings without the ith and jth characters is one shorter than their lcs with those characters
- in other words, whatever the lcs of XXXX and YYYYY is without the two R's, then the lcs with the R's is one more



Step 2

- case 2: $s[i] \neq t[j]$
- whatever the lcs of $s[0..i]$ and $t[0..j]$ is, it is not true that both $s[i]$ and $t[j]$ are part of it (why?)
- therefore, the lcs of $s[0..i]$ and $t[0..j]$ must be either
  - the lcs of $s[0..i-1]$ and $t[0..j]$ or
  - the lcs of $s[0..i]$ and $t[0..j-1]$

- now there is enough information to construct a recurrence relation that defines $\text{opt}(i, j)$

Step 2

$$\text{opt}(i,j) = \begin{cases} \text{opt}(i-1,j-1)+1 & \text{if } s[i] = t[j] \\ \max \begin{cases} \text{opt}(i-1,j) \\ \text{opt}(i,j-1) \end{cases} & \text{if } s[i] \neq t[j] \end{cases}$$

Base case: if $i = 0$ or $j = 0$, then $\text{opt}(i,j) = 0$

Step 3

- as before, put a dummy blank character onto the beginning of each string
- what is the shape of the memo table?
- what does an entry in the memo matrix represent?
- what is the minimum value in the memo matrix?

Step 3

```
1  size_t opt(size_t i, size_t j, Matrix<size_t>& memo,
2             const string& s, const string& t)
3  {
4    if (memo.at(i, j) == SIZE_MAX)
5    {
6      if (i == 0 || j == 0)
7      {
8        memo.at(i, j) = 0;
9      }
10     else if (s.at(i) == t.at(j))
11     {
12       memo.at(i, j) = opt(i - 1, j - 1, memo, s, t) + 1;
13     }
14     else
15     {
16       memo.at(i, j) = max(opt(i, j - 1, memo, s, t),
17                           opt(i - 1, j, memo, s, t));
18     }
19   }
20   return memo.at(i, j);
21 }
```

LCS

- build the LCS memo table for these two sequences:

G V C E K S T

G D V E G T A

Longest Common Subsequence

- an LCS matrix

		-	G	V	C	E	K	S	T
		0	1	2	3	4	5	6	7
-	0	0	0	0	0	0	0	0	0
G	1	0							
D	2	0							
V	3	0							
E	4	0							
G	5	0							
T	6	0							
A	7	0							

Longest Common Subsequence

- an LCS matrix

		-	G	V	C	E	K	S	T
		0	1	2	3	4	5	6	7
-	0	0	0	0	0	0	0	0	0
G	1	0	1	1	1	1	1	1	1
D	2	0							
V	3	0							
E	4	0							
G	5	0							
T	6	0							
A	7	0							

Longest Common Subsequence

- an LCS matrix

		-	G	V	C	E	K	S	T
		0	1	2	3	4	5	6	7
-	0	0	0	0	0	0	0	0	0
G	1	0	1	1	1	1	1	1	1
D	2	0	1	1	1	1	1	1	1
V	3	0	1	2	2	2	2	2	2
E	4	0	1	2	2	3	3	3	3
G	5	0	1	2	2	3	3	3	3
T	6	0	1	2	2	3	3	3	4
A	7	0	1	2	2	3	3	3	4

LCS Traceback

- the step 4 traceback is a little different
- consider the pattern of values in the matrix

		-	G	V	C	E	K	S	T
		0	1	2	3	4	5	6	7
-	0	0	0	0	0	0	0	0	0
G	1	0	1	1	1	1	1	1	1
D	2	0	1	1	1	1	1	1	1
V	3	0	1	2	2	2	2	2	2
E	4	0	1	2	2	3	3	3	3
G	5	0	1	2	2	3	3	3	3
T	6	0	1	2	2	3	3	3	4
A	7	0	1	2	2	3	3	3	4

LCS Traceback

- the traceback goes from the top-left corner of one rectangle diagonally up and over one

		-	G	V	C	E	K	S	T
		0	1	2	3	4	5	6	7
-	0	0	0	0	0	0	0	0	0
G	1	0	1	1	1	1	1	1	1
D	2	0	1	1	1	1	1	1	1
V	3	0	1	2	2	2	2	2	2
E	4	0	1	2	2	3	3	3	3
G	5	0	1	2	2	3	3	3	3
T	6	0	1	2	2	3	3	3	4
A	7	0	1	2	2	3	3	3	4