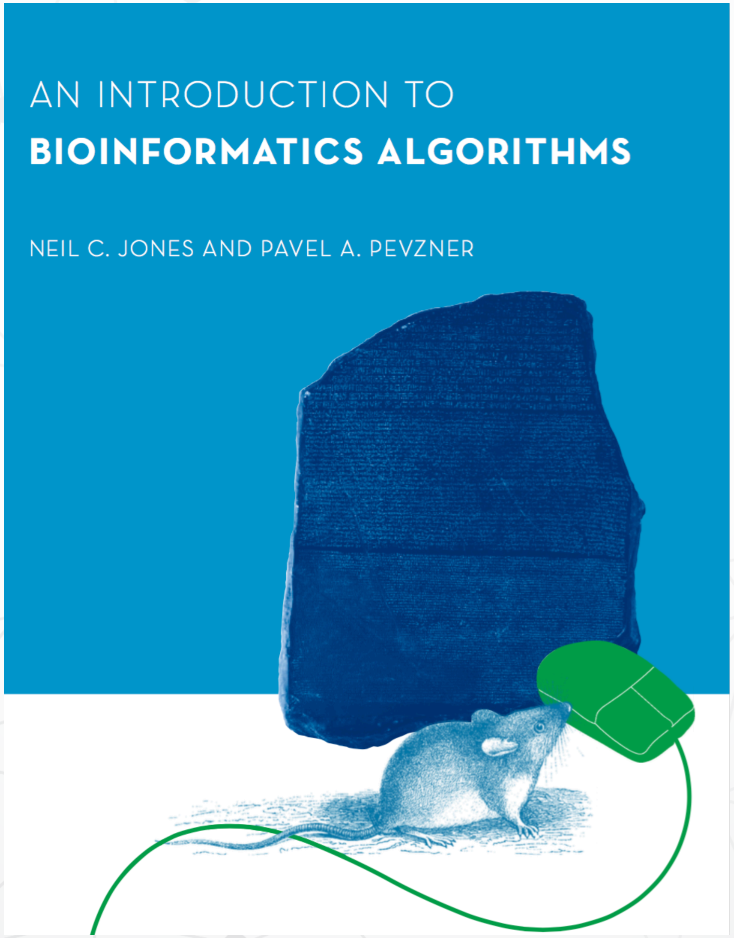
Bioinformatics Algorithms

by Eka Antonius Kurniawan

Outline

- Change Problem
- Brute Force Algorithm for Change Problem
- Recursive for Change Problem
- Greedy Algorithm for Change Problem
- Dynamic Programming for Change Problem
- Manhattan Tourist Problem
- Biological Equivalent of Manhattan Tourist Problem
- Dynamic Programming for Manhattan Tourist Problem

An Introduction to Bioinformatics Algorithms by Neil C. Jones and Pavel A. Pevzner



http://www.math-info.univ-paris5.fr/~lomn/Cours/BC/Publis/Complements/introductiontoBioinformaticsAlgorithms.pdf

Change Problem

Indonesian Change Problem

Convert some amount of money into the fewest number of coins.

Input: An amount of money (M)

Output: The fewest number of coins (C) that values equal to amount of money (M) where M = 1000r + 500l + 200d + 100s and C = r + l + d + s.

Terminology

$M = 1000r + 500l + 200d + 100s$

$M = c_1.i_1 + c_2.i_2 + c_3.i_3 + c_4.i_4$

$\mathbf{c} = (c_1, c_2, c_3, c_4)$

$C = r + l + d + s$

$C = i_1 + i_2 + i_3 + i_4$

$d = 4$

Brute Force Algorithm for Change Problem

Brute Force Algorithm

```
BruteForceChange(M,c,d)
    smallestNumberOfCoins ← ∞
    for each (i_1,...,i_d) from (0,...,0) to (M/c_1,...,M/c_d)
        valueOfCoins ← sum(c_k.i_k) where k from 1 to d
        if valueOfCoins = M
            numberOfCoins = sum(i_k) where k from 1 to d
            if numberofCoins < smallestNumberOfCoins
                smallestNumberOfCoins ← numberofCoins
                bestChange ← (i_1,...,i_d)
    return bestChange
```

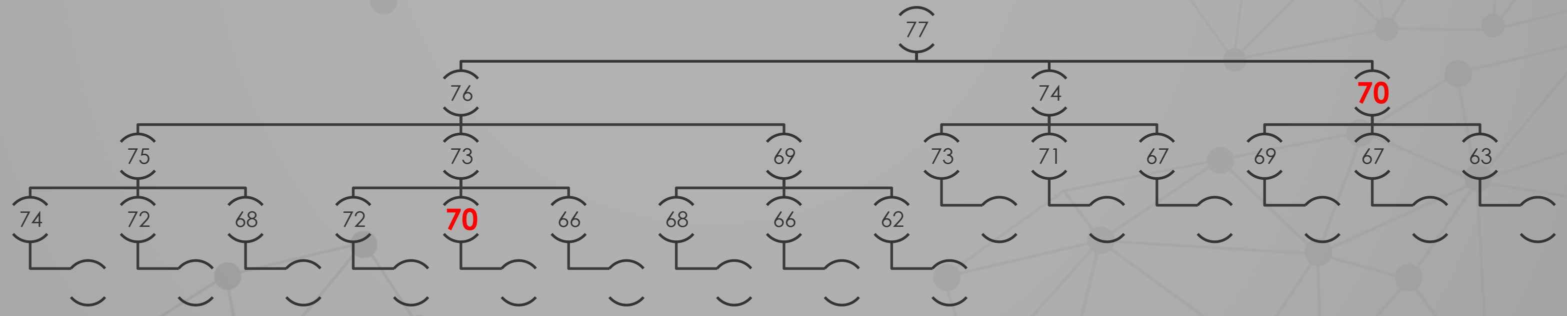
Recursive Algorithm for Change Problem

Recursive Algorithm

```
RC(M)
  if (M<0) return ∞
  else if (M=0) return 0
  else if (M=c_1) or ... (M=c_d) return 1
  else return min(RC(M-c_1)+1,..., RC(M-c_d)+1)
```

Recursive Algorithm Example

M=77
c=(1,3,7)
d=3



Greedy Algorithm for Change Problem

Greedy Algorithm

```
GreedyChange(M,c,d)
  C ← 0
  for c in each sort(c, descending)
    if M >= c
      i ← int(M/c)
      C ← C + i
      M ← M – (c.i)
  return C
```

Greedy Algorithm Example

M=60
c=(25,20,10,5,1)
change=(25,25,10)

M=40
c=(25,20,10,5,1)
change=(25,10,5); C=3
What about: **change**=(20,20); C=2

Dynamic Programming for Change Problem

Recurrence Relation

$$bestNumCoins_M = \min \begin{cases} bestNumCoins_{M-c_1} + 1 \\ bestNumCoins_{M-c_2} + 1 \\ \ldots \\ bestNumCoins_{M-c_d} + 1 \end{cases}$$

Dynamic Programming

DPChange(M,**c**,d)
    $bestNumCoins_0 \leftarrow 0$
    for m **from** 1 **to** M
        $bestNumCoins_m \leftarrow \infty$
        for i **from** 1 **to** d
            if $m >= c_i$
                if $bestNumCoins_{m-ci} + 1 < bestNumCoins_m$
                    $bestNumCoins_m \leftarrow bestNumCoins_{m-ci} + 1$

Dynamic Programming Example

M=9
c=(1,3,7)
d=3

0
0

0	1
0	1

0	1	2
0	1	2

0	1	2	3
0	1	2	1

0	1	2	3	4
0	1	2	1	2

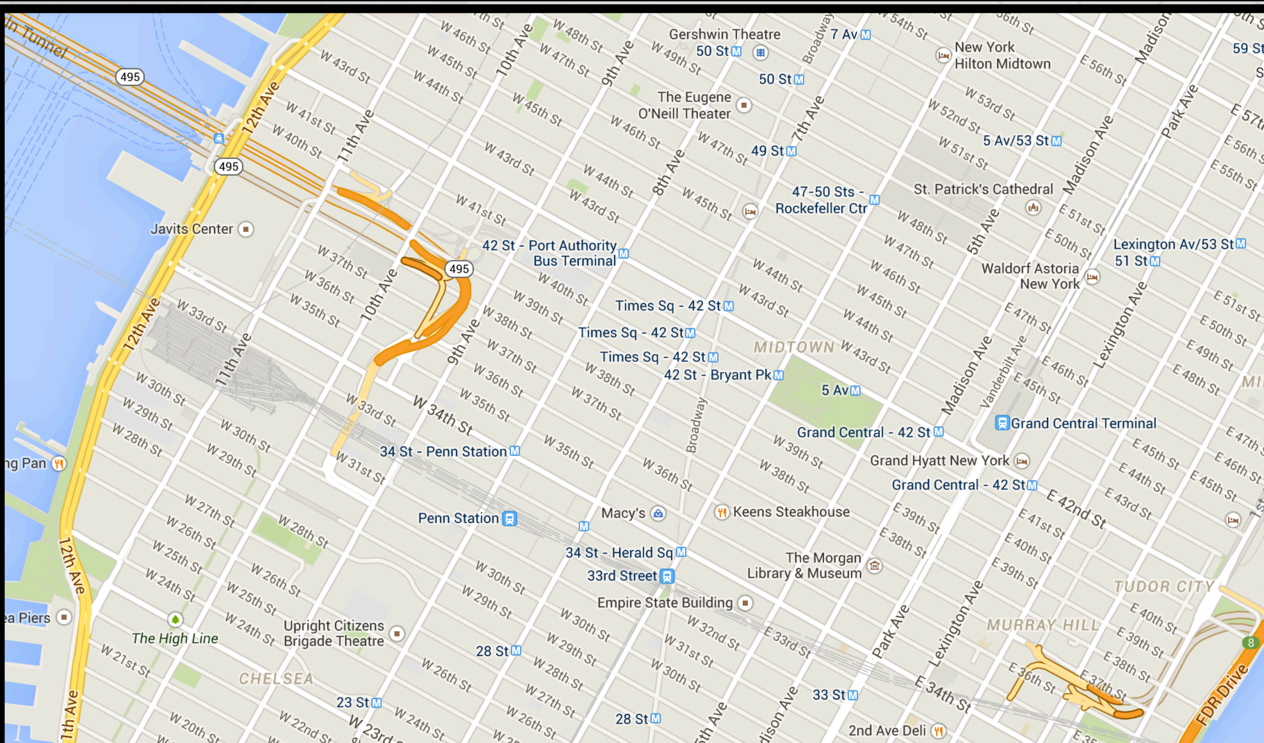
0	1	2	3	4	5
0	1	2	1	2	3

0	1	2	3	4	5	6
0	1	2	1	2	3	2

0	1	2	3	4	5	6	7
0	1	2	1	2	3	2	1

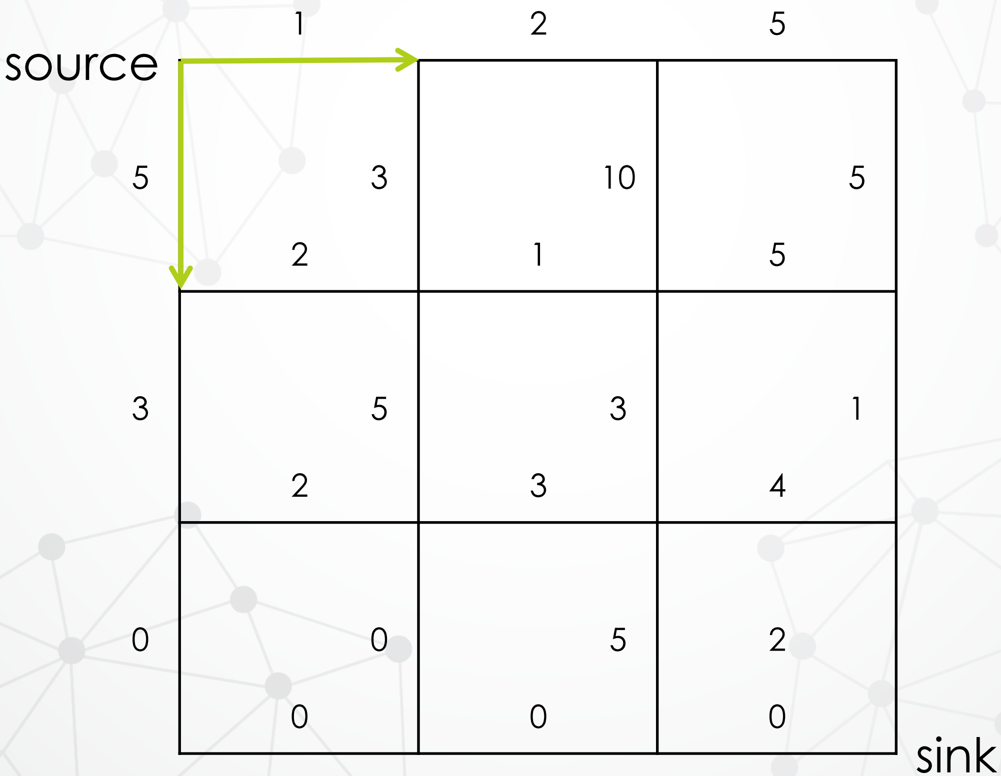
0	1	2	3	4	5	6	7	8
0	1	2	1	2	3	2	1	2

0	1	2	3	4	5	6	7	8	9
0	1	2	1	2	3	2	1	2	3

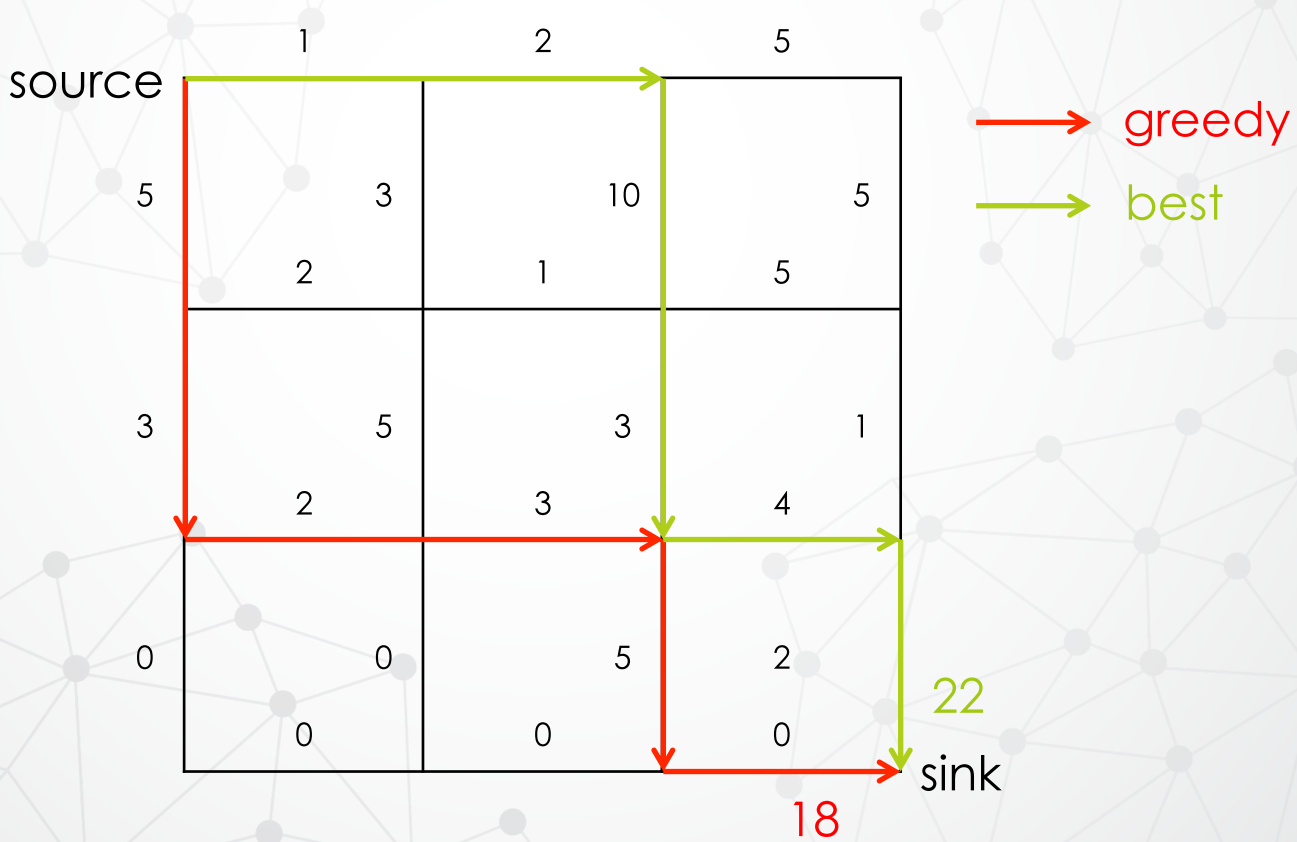


Manhattan Tourist Problem

Abstraction of Manhattan Tourist Problem



Greedy Algorithm for Manhattan Tourist Problem



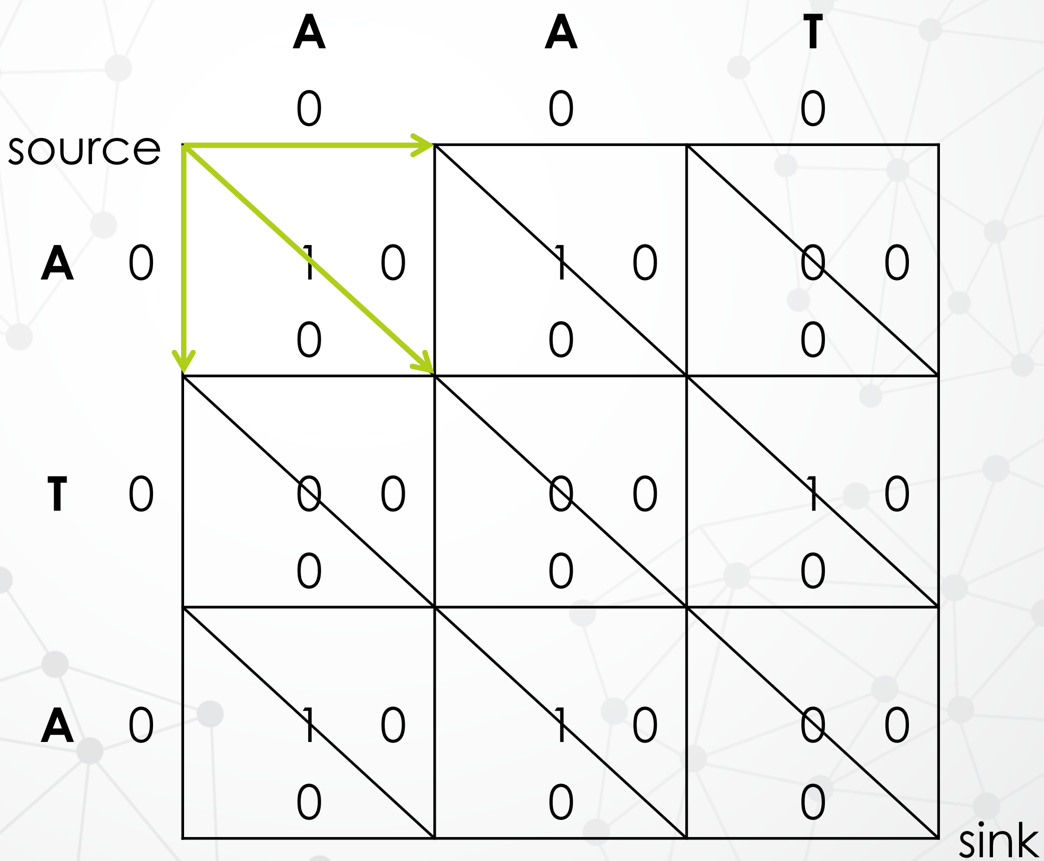
Biological Equivalent of Manhattan Tourist Problem

Biological Abstraction of Manhattan Tourist Problem

Input:

v = A T A

w = A A T



Dynamic Programming for Manhattan Tourist Problem

Recurrence Relation

$$s_{i,j} = \max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1 \text{; if } v_i = w_i \end{cases}$$

Dynamic Programming

```
DPSequence(v,w)
    for i from 0 to n
        s_{i,0} ← 0
    for j from 0 to m
        s_{0,j} ← 0
    for i from 0 to n
        for j from 0 to m
            s_{i,j} ← runRecurrenceRelation()
    return s_{i,j}
```

DP Example of Manhattan Tourist Problem

Score Table

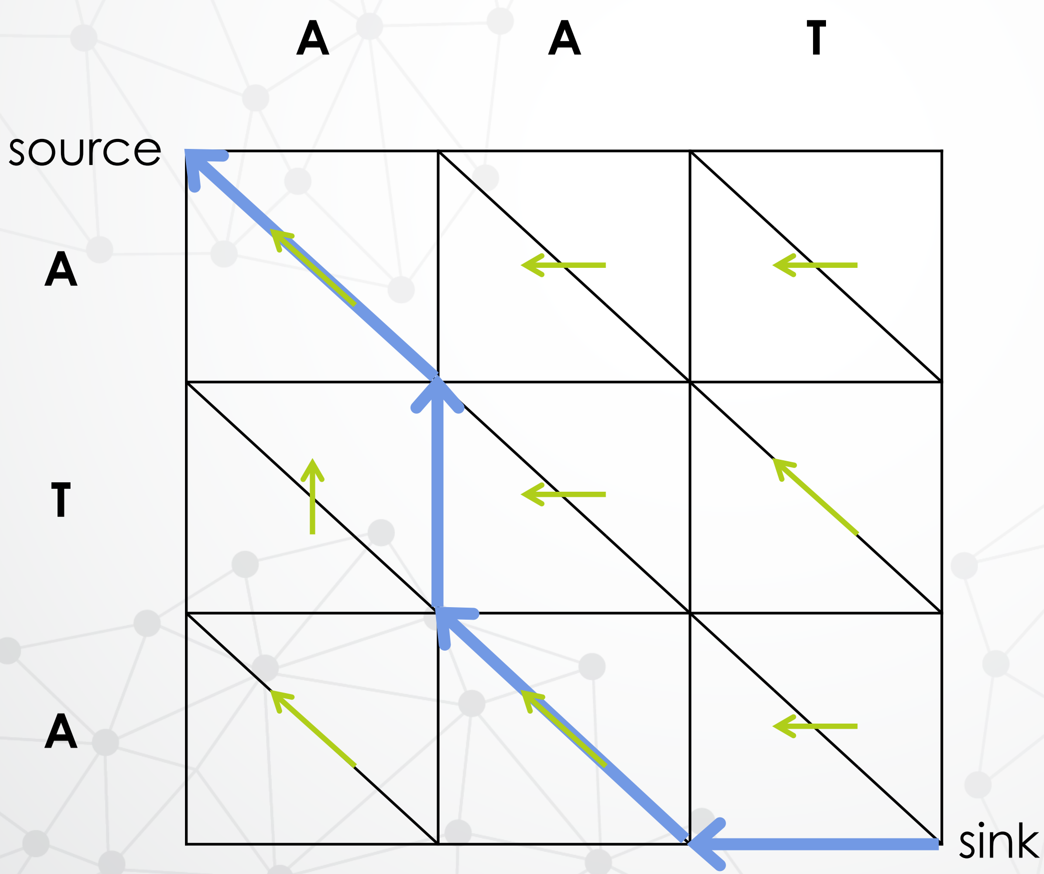
		A	**A**	**T**
		0	0	0
A	0	1 0 0 **1**	1 0 0 **1**	0 0 0 **1**
T	0	0 0 0 **1**	0 0 0 **1**	1 0 0 **2**
A	0	1 0 0 **1**	1 0 0 **2**	0 0 0 **2**

Direction Table

	A	**A**	**T**
A	↖	←	←
T	↑	←	↖
A	↖	↖	←

DP Example of Manhattan Tourist Problem

Direction Table



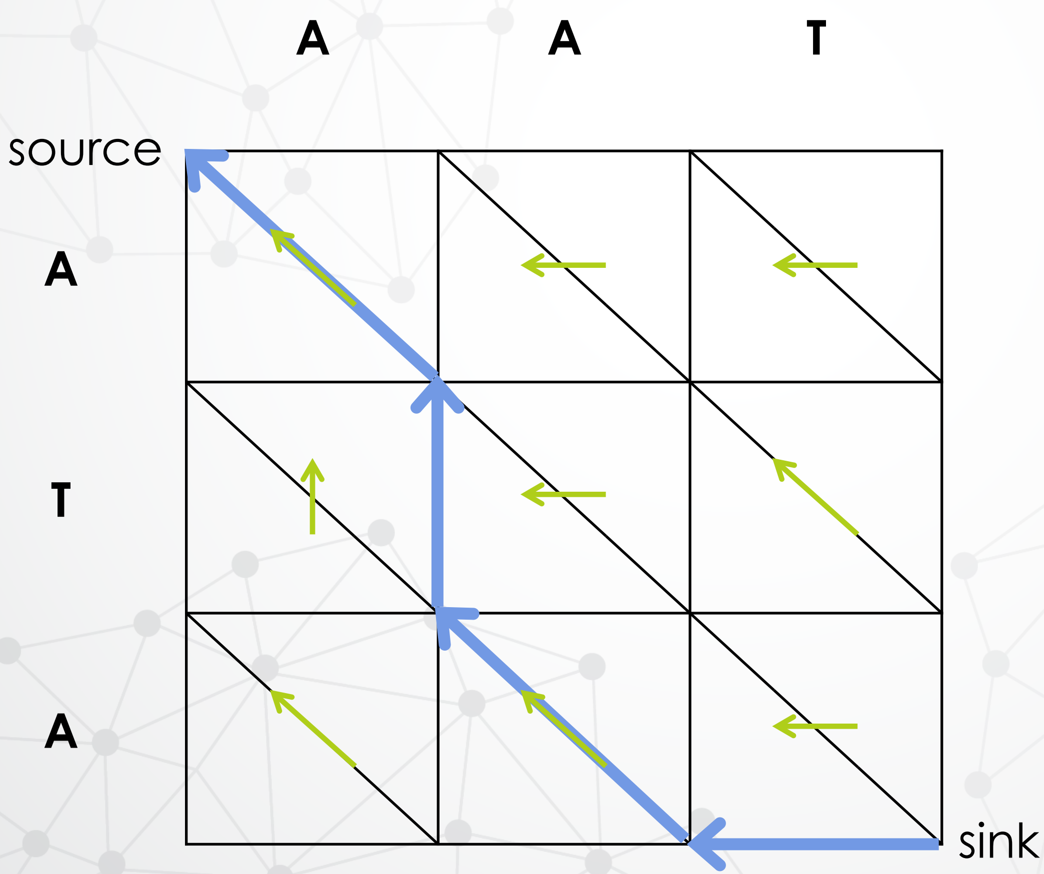
Output:

v = A T A –

w = A - A T

DP Example of Manhattan Tourist Problem

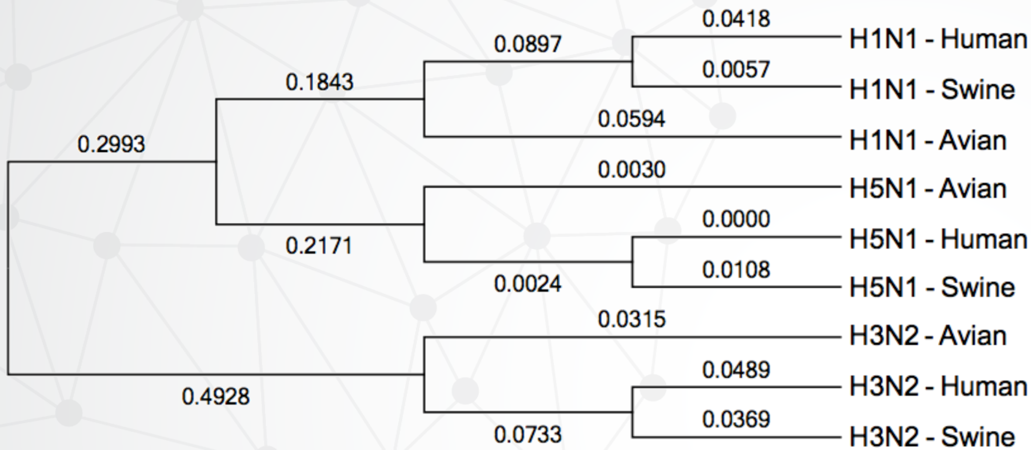
Direction Table

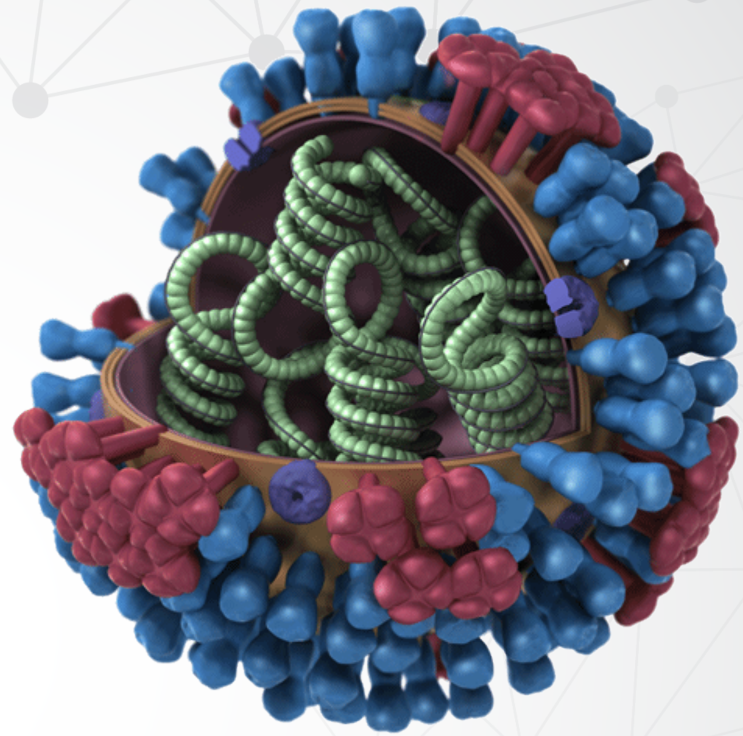


Output:

v = A T A –

w = A - A T





Triple-reassortant Influenza A Virus

http://www.cdc.gov/flu/images.html

Sequence Alignment of Influenza A Hemagglutinin Protein

Input:

V =

```
MEAKLFVLFCTFTVLKADTICVGYHANNSTDTVDTVLEKNVTVTHSVNLLEDSHNGKLCSLNGIAPLQLGKCNVAGWLLGNPE
CDLLLTANSWSYIIETSNSENGTCYPGEFIDYEELREQLSSVSSFEKFEIFPKANSWPNHETTKGVTAACSYSGASSFYRNLL
WITKKGTSYPKLSKSYTNNKGKEVLVLWGVHHPPTTSEQQSLYQNTDAYVSVGSSKYNRRFTPEIAARPKVRGQAGRMNYYWT
LLDQGDTITFEATGNLIAPWYAFALNKGSDSGIITSDAPVHNCDTRCQTPHGALNSSLPFQNVHPITIGECPKYVKSTKLRMA
TGLRNVPSIQSRGLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMNTQFTAVGKEFN
NLERRIENLNKKVDDGFLDVWTYNAELLVLLENERTLDFHDSNVRNLYEKVRSQLRNNAKELGNGCFEFYHKCDDECMESVKN
GTYDYPKYSEESKLNREEIDGVKLESMGVYQILAIYSTVASSLVLLVSLGAISFWMCSNGSLQCRICI
```

W =

```
MKAILVVLLYTFATANADTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGWILGNPE
CESLSTASSWSYIVETSSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKTSSWPNHDSNKGVTAACPHAGAKSFYKNLI
WLVKKGNSYPKLSKSYINDKGKEVLVLWGIHHPSTSADQQSLYQNADAYVFVGTSRYSKKFKPEIAIRPKVRDQEGRMNYYWT
LVEPGDKITFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITIGKCPKYVKSTKLRLA
TGLRNVPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSGYAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFN
HLEKRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVKN
GTYDYPKYSEEAKLNREEIDGVKLESTRIYQILAIYSTVASSLVLVVSLGAISFWMCSNGSLQCRICI
```

Aligned Sequences of Influenza A Hemagglutinin Protein

Output:

```
vp = M-EA-KLFV-L--FCTF-TVLKA--DT--ICVGYHANNSTDTVDTVLEKNVTVTHSVNLLED-SHNGKLC-SL-NG-
wp = MK-AI-L-VVLLY--TFAT---ANADTLCI--GYHANNSTDTVDTVLEKNVTVTHSVNLLEDK-HNGKLCK-LR-GV

IAPL-QLGKCN-VAGW-LLGNPEC--DL-LLTANS-WSYI-IETSNS--ENGTCYPG-EFIDYEELREQLSSVSSFE-KFEI
-APLH-LGKCNI-AGWIL-GNPECES-LS--TA-SSWSYIV-ETS-SSD-NGTCYPGD-FIDYEELREQLSSVSSFER-FEI

FPK-ANS-WPNH---ETTKGVTAAC---SYSGA-SSFY-RNL--L-WITKKG-TSYPKLSKSY-TN-NKGKEVLVLWG-VHH
FPKT--SSWPNHDSN---KGVTAACPHA---GAKS-FYK-NLIWLV---KKGN-SYPKLSKSYI-ND-KGKEVLVLWGI-HH

P-PTTS--EQQSLYQN-TDAYV-SVG-S--SK-YNRRF-TPEIA-ARPKVR-GQ-AGRMNYYWTL---LDQGD-TITFEATG
PS-T-SAD-QQSLYQNA-DAYVF-VGTSRYSKK----FK-PEIAI-RPKVRD-QE-GRMNYYWTLVEP---GDK-ITFEATG

NL--IAP-WYAFA---LN-KGSDSGII-TSD-APVH--NCDT-RCQTP-HGA-LN-SSLPFQN-VHPITIG-ECPKYVKSTK
NLVV--PR-YAFAMER-NA-GS--GIII-SDT-PVHDCN--TT-CQTPK-GAI-NTS-LPFQNI-HPITIGK-CPKYVKSTK

LR-MATGLRNVPSIQSRGLFGAIAGFIEGGWTGM-IDGWYGYHHQNEQGSGYAAD-QKSTQNAID-GITNKVNSVIEKMNTQ
LRL-ATGLRNVPSIQSRGLFGAIAGFIEGGWTGMV-DGWYGYHHQNEQGSGYAADL-KSTQNAIDE-ITNKVNSVIEKMNTQ

FTAVGKEFN-NLE-RRIENLNKKVDDGFLD-VWTYNAELLVLLENERTLD-FHDSNV-RNLYEKVRSQL-RNNAKE-LGNGC
FTAVGKEFNH-LEKR-IENLNKKVDDGFLDI-WTYNAELLVLLENERTLDY-HDSNVK-NLYEKVRSQLK-NNAKEI-GNGC

FEFYHKCD--DECMESVKNGTYDYPKYSEE-SKLNREEIDGVKLES---MGVYQILAIYSTVASSLVLLV-SLGAISFWMCS
FEFYHKCDNT--CMESVKNGTYDYPKYSEEA-KLNREEIDGVKLESTRI---YQILAIYSTVASSLVL-VVSLGAISFWMCS

NGSLQCRICI
NGSLQCRICI
```

Demo

http://nbviewer.ipython.org/github/ekaakurniawan/umn-kapita_selecta-2015-ganjil/blob/master/W5%20-%20Bioinformatics%20Algorithms/DynamicProgramming.ipynb

Q&A