Kieran Finnegan

Genetic Algorithm Assignment

Learning Abstract:

The goal of this project was to identify a problem that needed solving using the evolutionary approach. By writing the solution in CLOS I had to faithfully mimic the process that was established during the development of the RBG-string GA.

The Tasks:

Task 1 The String:

Demo:

```
CL-USER> *cities*
(A B C D E F G H I J K L)
CL-USER> ( tsp-string )
(C J A E L I H D B F K G)
CL-USER> ( tsp-string )
(J L D I A H F G C B K E)
CL-USER> ( tsp-string )
(C G H E J A D F K I L B)
```

Code:

```
( defun tsp-string-builder ( lst &aux ele rem )
```

```
  ( cond
    ( ( null lst ) nil )
          ( t
            ( setf ele ( pick lst ) )
            ( setf rem ( remove ele lst :count 1 ) )
            ( snoc ele ( tsp-string-builder rem ) )
          )
  )
)

( defun tsp-string()
  ( tsp-string-builder *cities* )
)
```

Task 2 Mutation pt 1:

Demo:

```
CL-USER> ( setf tour '( 1 2 3 4 5 ) )
(1 2 3 4 5)
CL-USER> ( mutation tour )
(1 2 5 4 3)
CL-USER> ( mutation tour )
(1 2 3 4 5)
CL-USER> ( mutation tour )
(1 2 4 3 5)
CL-USER> ( mutation tour )
(1 4 2 3 5)
CL-USER> ( mutation s )
(1 2 3 4 4 3 2 1)
CL-USER> ( mutation s )
(1 2 4 4 3 3 2 1)
CL-USER> ( mutation s )
(1 3 4 4 2 3 2 1)
CL-USER> ( setf x ( tsp-string ) )
(F G H D L A C J E I B K)
CL-USER> ( mutation x )
(F G K D L A C J E I B H)
CL-USER>  ( mutation x )
(F G K D L A C J I E B H)
CL-USER>
```

Code:

```
( defun swap ( lst n1 n2 &aux city1 city2 )
  ( setf city1 ( nth n1 lst ) )
  ( setf city2 ( nth n2 lst ) )
  ( setf ( nth n1 lst ) city2 )
  ( setf ( nth n2 lst ) city1 )
  lst
)

( defun mutation ( tour &aux n1 n2 )
  ( setf n1 ( random ( length tour ) ) )
  ( setf n2 ( random ( length tour ) ) )
  ( swap tour n1 n2 )
)
```

Task 3 Crossover pt 1:

Demo:

```
CL-USER> ( setf m ( tsp-string ) )
(C G I F E B A H K D L J)
CL-USER> ( setf f ( tsp-string ) )
(L K J F C D B E H A I G)
CL-USER> ( crossover m f )
(C G I F B E H A K D L J)
CL-USER> ( crossover m f )
(C G I F B E H A K D L J)
CL-USER> m
(C G I F E B A H K D L J)
CL-USER> f
(L K J F C D B E H A I G)
CL-USER>
```

Code:

```
( defmethod crossover ( (m list) (f list) )
  ( setf span-5 (common-span-n m f 5) )
  ( setf span-4 (common-span-n m f 4) )
  ( setf span-3 (common-span-n m f 3) )
  ( cond
    ( (not (null span-5))
      ( setf span-m (nth 0 span-5) )
      ( setf span-f (nth 1 span-5) )
```

```
      ( swap-lst m span-m span-f )
    )
    ( (not (null span-4))
      ( setf span-m (nth 0 span-4) )
      ( setf span-f (nth 1 span-4) )
      ( swap-lst m span-m span-f )
    )
    ( (not (null span-3))
      ( setf span-m (nth 0 span-3) )
      ( setf span-f (nth 1 span-3) )
      ( swap-lst m span-m span-f )
    )
    ( t
      nil
    )
  )
)

( setf *index* 0 )

( defmethod swap-lst (m span-m span-f)
  ( setf m (remove-span span-m m 0) )
  ( add-span span-f m 0 )
)

( defmethod add-span (span lst n)
  ( cond
    ( (= n (+ 1 (length lst)))
      nil
    )
    ( (= n *index* )
      ( setf *index* 0 )
      ( append (subseq lst 0 n) span (subseq lst n (length lst)) )
    )
    ( t
      ( add-span span lst (+ 1 n) )
    )
  )
)
```

```
( defmethod remove-span (span lst n)
  ( cond
    ( (null span)
      ( setf *index* n )
      lst
    )
    ( (eq (car span) (nth n lst))
      ( remove-span (cdr span) (remove (nth n lst) lst) n )
    )
    ( t
      ( remove-span span lst (+ 1 n) )
    )
  )
)

( defmethod common-span-n ( mother father n )
  ( equal-set-1 mother father 0 n 0 n )
)

( defmethod equal-set-1 ( m f mn1 mn2 fn1 fn2 &aux set-m set-f )
  ( cond
    ( (= mn2 (+ 1 (length m)))
      nil
    )
    ( t
      ( setf set-m (subseq m mn1 mn2) )
      ( setf set-f (subseq f fn1 fn2) )
      ( cond
        ( (eq ( equal-set-2 m f mn1 mn2 fn1 fn2 ) nil)
          ( equal-set-1 m f (+ mn1 1) (+ mn2 1) fn1 fn2 )
        )
        ( t
          ( equal-set-2 m f mn1 mn2 fn1 fn2 )
        )
      )
    )
  )
)

( defmethod equal-set-2 ( m f mn1 mn2 fn1 fn2 &aux set-m set-f )
```

```
  ( cond
    ( (= fn2 (+ 1 (length f)))
      nil
    )
    ( t
      ( setf set-m (subseq m mn1 mn2) )
      ( setf set-f (subseq f fn1 fn2) )
      ( cond
        ( (not (eq (set-difference set-m set-f) nil))
          ( equal-set-2 m f mn1 mn2 (+ fn1 1) (+ fn2 1) )
        )
        ( t
          ( list set-m set-f )
        )
      )
    )
  )
)
```

Task 4 Fitness:

Demo:

CL-USER> (mutation-demo)
s = (L A J K E B H G I F C D)
m = (L A J K D B H G I F C E)

s = (L A J K D B H G I F C E)
m = (L A C K D B H G I F J E)

s = (L A C K D B H G I F J E)
m = (L A C K H B D G I F J E)

s = (L A C K H B D G I F J E)
m = (L A C K H B D G I F J E)

s = (L A C K H B D G I F J E)
m = (L A C K J B D G I F H E)

s = (L A C K J B D G I F H E)
m = (L A C H J B D G I F K E)

s = (L A C H J B D G I F K E)
m = (L A C H J G D B I F K E)

s = (L A C H J G D B I F K E)
m = (L F C H J G D B I A K E)

s = (L F C H J G D B I A K E)

```
m = (L F C H J G D B I A K E)

s = (L F C H J G D B I A K E)
m = (L F C H J B D G I A K E)

NIL
CL-USER> ( crossover-demo )
m = (K J G A C F B E L H I D)
x = (K J G A C F B E L H I D)
f = (I D K C H A J L G F B E)

NIL
```

Code:

```
( defmethod mutation-demo ()
  ( setf s (tsp-string) )
  ( dotimes (i 10)
    ( format t "s = ~A~%" s )
    ( setf m ( mutation s ) )
    ( format t "m = ~A~%~%" m )
  )
)


( defmethod crossover-demo ()
  ( setf m (tsp-string) )
  ( setf f (tsp-string) )
  ( setf x (crossover m f) )
  ( cond
    ( (null x)
      ( format t "m = ~A~%" m )
      ( format t "x = ~A~%" x )
      ( format t "f = ~A~%" f )
      ( format t "Crossover not applicable. Reshuffling... ~%~%" )
      ( crossover-demo )
    )
    ( t
      ( format t "m = ~A~%" m )
      ( format t "x = ~A~%" x )
      ( format t "f = ~A~%~%" f )
```

```
    )
  )
)
```

Task 5 Fitness:

Demo:

```
CL-USER> ( setf x ( tsp-string ) )
(K J I B G A F H L E C D)
CL-USER> ( setf y ( tsp-string ) )
(F J I K D H G C L E B A)
CL-USER> ( setf z ( tsp-string ) )
(H J K E C D A G F B I L)
CL-USER> ( fitness-distance x )
2267
CL-USER> ( fitness-distance y )
2159
CL-USER> ( fitness-distance z )
2070
```

Code:

```
( defmethod fitness-distance ( ( p-list list ) )
  ( if ( null ( nth 1 p-list ) )
    0
```

```
    ( + ( get ( car p-list ) ( nth 1 p-list ) ) ( fitness-distance ( cdr p-list ) ) )
  )
)


( defmethod fitness-demo ( &aux x fitness )
  ( setf x ( tsp-string ) )
  ( format t "Directly applying the fitness metrics ...~%" )
  ( format t "fitness-distance = ~A~%" ( fitness-distance x ) )
)
```

Task 6 The indivdual:

Demo:

```
CL-USER> ( setf tsp ( tsp-string ) )
(F J E D C G I H K A B L)
CL-USER> ( setf *fitness* #'fitness-distance )
#<STANDARD-GENERIC-FUNCTION COMMON-LISP-USER::FITNESS-DISTANCE
(2)>
CL-USER> ( setf tsp-i ( new-individual 1 tsp ) )
#<INDIVIDUAL {1002BD8B23}>
CL-USER> ( idividual-number tsp-i )
{1002C47DC3}>.
CL-USER> ( individual-number tsp-i )
1
CL-USER> ( individual-tsp-string tsp-i )
(F J E D C G I H K A B L)
CL-USER> ( display tsp-i )
1       (F J E D C G I H K A B L)    2576
NIL
CL-USER> ( funcall *fitness* tsp )
2576
CL-USER> ( setf r ( random-individual ) )
#<INDIVIDUAL {10030B4223}>
CL-USER> ( display r )
```

```
0      (H K F C L D I B G J E A)    2167
NIL
CL-USER> ( setf r ( random-individual ) )
#<INDIVIDUAL {10031083A3}>
CL-USER> ( display r )
0      (B K E L D I F A J H G C)    1600
NIL
CL-USER> ( individual-demo )
0      (G I F B L K E J H C A D)    1645
1      (I L D C E J A G F B K H)    2279
2      (C A L I K E J H F G D B)    1797
3      (A L J H F I G D C E B K)    2342
Fitness of i0 = 1645
Fitness of i1 = 2279
Fitness of i2 = 1797
Fitness of i3 = 2342
NIL
```

Code:

```
( defclass individual()
  (
    ( tsp-string :accessor individual-tsp-string :initarg :tsp-string )
    ( fitness :accessor individual-fitness :initarg :fitness )
    ( number :accessor individual-number :initarg :number )
  )
)

( defmethod random-individual ( &aux tsp )
  ( setf tsp ( tsp-string ) )
  ( make-instance 'individual
    :tsp-string tsp
    :fitness ( funcall *fitness* tsp )
    :number 0
  )
)

( defmethod new-individual ( ( nr number )( notes list ) )
  ( make-instance 'individual
    :tsp-string notes
    :fitness (funcall *fitness* notes)
```

```
    :number nr
  )
)

( defmethod display ( ( i individual ) )
  ( display-nnl i ) ( terpri )
)

( defmethod display-nnl ( ( i individual ) )
  ( prin1 ( individual-number i ) )
  ( princ ( filler ( individual-number i ) ) )
  ( prin1 ( individual-tsp-string i ) )
  ( princ "    " )
  ( prin1 ( individual-fitness i ) )
  ( princ ( filler (individual-fitness i ) ) )
)

( defmethod filler (( n number ))
  ( cond
    ( ( < n 10 ) "       " )
    ( ( < n 100 ) "      " )
    ( ( < n 1000 ) "     " )
    ( ( < n 10000 ) "    " )
    ( ( < n 100000 ) "   " )
  )
)

( defmethod fitness-distance ( ( i individual ) )
  ( fitness-distance ( individual-tsp-string i ) )
)

( defmethod individual-demo (&aux i0 i1 i2 i3 one two three)
  ( setf *fitness* #'fitness-distance )
  ( setf i0 ( random-individual ) )
  ( display i0 )
  ( setf one ( tsp-string ) )
  ( setf i1 ( new-individual 1 one ) )
  ( display i1 )
  ( setf two ( tsp-string ) )
  ( setf i2 ( new-individual 2 two ) )
  ( display i2 )
  ( setf three ( tsp-string ) )
  ( setf i3 ( new-individual 3 three ) )
  ( display i3 )
```

```
  ( format t "Fitness of i0 = ~A~%" ( funcall *fitness* (individual-tsp-string i0 ) ) )
  ( format t "Fitness of i1 = ~A~%" ( funcall *fitness* (individual-tsp-string i1 ) ) )
  ( format t "Fitness of i2 = ~A~%" ( funcall *fitness* (individual-tsp-string i2 ) ) )
  ( format t "Fitness of i3 = ~A~%" ( funcall *fitness* (individual-tsp-string i3 ) ) )
  nil
)
```

Task 7 Population:

Demo:

```
CL-USER> ( setf p ( initial-population ) )
;   caught 1 WARNING condition
#<POPULATION {100323E1F3}>
CL-USER> ( display p )


Generation 0 Population ...

1       (L B H C J K A F G I E D)    2018
2       (E J B A D F H G L K I C)    2327
3       (K E B H A F D I C G L J)    1549
4       (L B K H J E C G A D I F)    2242
5       (E K F C D H B I G A J L)    2041
6       (G D B E C F K I H A J L)    2124
7       (L B H D A J F I G K E C)    2214
8       (F H K C L J B I E G D A)    2531
9       (J E I L C A D G K B F H)    2102
10      (E C A D B L G H F I K J)    1883
11      (F B C G A L E H K I J D)    2419
12      (F G K J B C L E H A I D)    2575
13      (E H J G F A L D C I B K)    2174
14      (K L F I D H E J G B A C)    2136
15      (G K H A F B L D E J C I)    2156
```

16 (D L A F I H G C E J B K) 2225
17 (F L J H K D B G C A E I) 2093
18 (L J B C A H F I E D G K) 1990
19 (E L C D H F A I B K G J) 2240
20 (F A I B J L K D C G E H) 2354
21 (F H B C J E G D L A I K) 2656
22 (B G H D E F C K I L J A) 2223
23 (K G A H E I C B D J L F) 2285
24 (I K F A C B L G J H E D) 2000
25 (E D H K B L I A F G C J) 2017
26 (G H L D F B A K I E C J) 2280
27 (B A C E K D H F L I J G) 1803
28 (K L D E G H J I F A C B) 1773
29 (E A F B K L G D H J I C) 2268
30 (K J F A L C B D I H G E) 2420
31 (G L C A I B H D F J K E) 1966
32 (G I J D B E F A H C K L) 1758
33 (F E D H B J A C G L I K) 1992
34 (L J K H C I D G F E A B) 2570
35 (F A I H J C K E G L D B) 2215
36 (C A I D K E B L H F G J) 1546
37 (A D K G E C B H L I F J) 2457
38 (D C A J F K I H B E L G) 2257
39 (B L D J C I H F A E K G) 2207
40 (B C E D A I F H L J G K) 2639
41 (I B D F J K L C A G E H) 2140
42 (J K I A B C G L F H D E) 2447
43 (E K A H J F I L B C D G) 1890
44 (L H K I J G F A D E C B) 2476
45 (H L F G E D B J K C A I) 2263
46 (F H D I J K L B G A C E) 1928
47 (L B A C K H G J E F I D) 2038
48 (H B J K E A L D F C G I) 2149
49 (L E B C F K A H I G J D) 2142
50 (E G A I D H J F C K L B) 2283
51 (J I G E K B C F L H D A) 2132
52 (B F I E C K D L A H J G) 2115
53 (A B H J E I G C D F K L) 2144
54 (K J F H B I G C D L E A) 2247
55 (L K D B I E G A C F J H) 1833

56	(K I H B J A L C E G D F)	2683
57	(D G K E H F J A I L C B)	2580
58	(B H J D A F L I E C G K)	2144
59	(G K H I F D B E C L A J)	2516
60	(L G E A C K I D H J B F)	2310
61	(K B C H F J A L D E I G)	2110
62	(H K I C G B J A L E F D)	2394
63	(C E A D B I L H J G F K)	2106
64	(L G E K H I C J B D A F)	2382
65	(L I A D F J G B H C K E)	1852
66	(E D L A K B C J I H G F)	2182
67	(H E K G J I A L C F B D)	2242
68	(K F A B J D E L G C I H)	2224
69	(B A F C I G E K L J H D)	2389
70	(A B H G D I E F J L C K)	2180
71	(F C I K L J G B A H D E)	2219
72	(C G E F H B L I J D K A)	2019
73	(C I A B H J G L D F E K)	1923
74	(K I J L H E C G B F A D)	2432
75	(E B H J L D G I F A K C)	2040
76	(I E G L C K A B D H F J)	2623
77	(L B K I E D C H A G F J)	1723
78	(H D I G L J K C B F E A)	2588
79	(E L C J D A I K B G H F)	2335
80	(J C F E D K G I B A H L)	2108
81	(K A C D E H I B J F L G)	2219
82	(G A I K B D F E H C L J)	2154
83	(C K J H L E G D A I B F)	2547
84	(G A I L C E K D H J F B)	2277
85	(B C D A J H E F I K L G)	2599
86	(J K D B H E L F C I A G)	2238
87	(K G E I D F J H A L B C)	2127
88	(G B C K E L F H D J I A)	2206
89	(A K D F H L J C I G B E)	2363
90	(A E H J K D F L G C B I)	2358
91	(C B D I J H F K E G L A)	1929
92	(K J E L C A I H F G B D)	2314
93	(L K D G C I J F A E B H)	2139
94	(G L F J H D K I A B C E)	2676
95	(I D K B L G A J C H E F)	1973

```
96      (A E F J K D B G I L C H)    2189
97      (A I F H B J E G D L C K)    2537
98      (G D H E J A C L I B K F)    2207
99      (K I H B G J L F E D C A)    2024
100     (C L A I F H E J K B D G)    2674

NIL
CL-USER> ( average p )
2210.06
CL-USER> ( select-individual p )
#<INDIVIDUAL {10031A0353}>
CL-USER> ( display ( select-individual p ) )
36      (C A I D K E B L H F G J)    1546
NIL
CL-USER> ( display ( select-individual p ) )
73      (C I A B H J G L D F E K)    1923
NIL
CL-USER> ( display ( select-individual p ) )
91      (C B D I J H F K E G L A)    1929
NIL
CL-USER> ( setf *select-demo* t )
T
CL-USER> ( display ( select-indivdual p ) )
{1003528B93}>.
CL-USER> ( display ( select-individual p ) )
the sample of individuals ...
42      (J K I A B C G L F H D E)    2447
28      (K L D E G H J I F A C B)    1773
73      (C I A B H J G L D F E K)    1923
97      (A I F H B J E G D L C K)    2537
65      (L I A D F J G B H C K E)    1852
45      (H L F G E D B J K C A I)    2263
66      (E D L A K B C J I H G F)    2182
59      (G K H I F D B E C L A J)    2516

the most fit of the sample...
28      (K L D E G H J I F A C B)    1773

28      (K L D E G H J I F A C B)    1773
NIL
```

```
CL-USER> ( display ( select-individual p ) )
the sample of individuals ...
2      (E J B A D F H G L K I C)    2327
66     (E D L A K B C J I H G F)    2182
97     (A I F H B J E G D L C K)    2537
22     (B G H D E F C K I L J A)    2223
51     (J I G E K B C F L H D A)    2132
80     (J C F E D K G I B A H L)    2108
63     (C E A D B I L H J G F K)    2106
44     (L H K I J G F A D E C B)    2476

the most fit of the sample...
63     (C E A D B I L H J G F K)    2106

63     (C E A D B I L H J G F K)    2106
NIL
CL-USER> ( display ( select-individual p ) )
the sample of individuals ...
77     (L B K I E D C H A G F J)    1723
94     (G L F J H D K I A B C E)    2676
3      (K E B H A F D I C G L J)    1549
42     (J K I A B C G L F H D E)    2447
97     (A I F H B J E G D L C K)    2537
8      (F H K C L J B I E G D A)    2531
21     (F H B C J E G D L A I K)    2656
19     (E L C D H F A I B K G J)    2240

the most fit of the sample...
3      (K E B H A F D I C G L J)    1549

3      (K E B H A F D I C G L J)    1549
NIL
CL-USER> ( population-demo )


Generation 0 Population ...

1      (K A F G C I H J E L D B)    2194
2      (D H C F G I L A E J B K)    2242
3      (E A L F D K C B J H G I)    2325
```

4	(J F E K H I A L B C G D)	2297
5	(K I B G L E D H F C A J)	1841
6	(G F I H J K E L D B C A)	1781
7	(G I D B H E K L A F C J)	1874
8	(B G J L C I E A D H F K)	2525
9	(I H J F K C D A L G E B)	2660
10	(D G J B H C E K F I A L)	1984
11	(G E D K J F A L I C H B)	2428
12	(K D E G A B I F C H J L)	1939
13	(I D J A L K F G E C B H)	2359
14	(J B L F D H I G K E C A)	1834
15	(L F J D G H K E I A C B)	2102
16	(L E I J F K G C B D H A)	2035
17	(I J L F B G H E D A K C)	2311
18	(A J H F C G I K D L B E)	1787
19	(B I L E G H K C F J D A)	2243
20	(H I E G D B A K L F J C)	2432
21	(J I B H K E C G A F L D)	1633
22	(J G E I A H D B K F C L)	2101
23	(K B I D H C F L A E J G)	2044
24	(D L C H K F G B J I E A)	2127
25	(J D G F L K A H C B I E)	1805
26	(A G J C K F D E B L H I)	1995
27	(F K H B A G E J L I D C)	2455
28	(E C J D A K G F I B H L)	2430
29	(D B C I G E H F A J K L)	2674
30	(G K A F C J E H B L D I)	2403
31	(F I H K G B D J L E C A)	2249
32	(C E G F J I L H D K B A)	2414
33	(D E G H L J B K A I C F)	2570
34	(G E A I C J B H F K L D)	2394
35	(B G I J D E H K F L A C)	2178
36	(L G F C I E A D K J B H)	2449
37	(F A L H G C I K J B D E)	2255
38	(B F L E D C G H A I J K)	1908
39	(F G J D K B L I E A C H)	1626
40	(B J K A I L H D E F G C)	2434
41	(H K D J F I B L E A C G)	1915
42	(E F G A K H I C L J D B)	2526
43	(K A C D G B L J F I H E)	2280

44	(G C I B E K L F H A D J)	1695
45	(A G D E J H B F I C L K)	2327
46	(C A H G D I L F B E J K)	1619
47	(D G I K L B H E J F C A)	2044
48	(F B J H I L D E A K C G)	2250
49	(D A C F J H B K E G L I)	2006
50	(J B H C L A K D G F E I)	2094
51	(J I E H B D C G A F K L)	2081
52	(K F G I C E B J D L H A)	2085
53	(E G K I H D B C F L J A)	2483
54	(C B F H A E G L K I D J)	2154
55	(H I K G L A J D C E B F)	2632
56	(E F B L I H C A D G K J)	1827
57	(D H I C L K G B E A J F)	2840
58	(B L E F H G J K I A D C)	2204
59	(B H E I L F C K G J D A)	2262
60	(F C A J G E K B H I L D)	1626
61	(A D B C L G J E F H K I)	2450
62	(C E K I B A H J D L F G)	1824
63	(H G J A F C E L K D B I)	1852
64	(C K B E H D I L J A F G)	2305
65	(D E B I A K C F G L H J)	2320
66	(A E I K J H F G C B D L)	2239
67	(D H C I A J E G F L B K)	2320
68	(L D H I G C B F J A K E)	2218
69	(H G I D A K L B J E C F)	2218
70	(L B I H J G C F E D K A)	1888
71	(J A C B G I E F D K H L)	2204
72	(D J G F C B A E H I L K)	2490
73	(L J A E F K B H D C G I)	2334
74	(F J I C G B A H L K D E)	2149
75	(K L E H C B G I F A D J)	2170
76	(F J I E K A B C L D H G)	2253
77	(K C L E H B I F J G D A)	2421
78	(B C K H G A E L D I J F)	2250
79	(E C L A H J F B D I K G)	2267
80	(E G H C I D F J L A K B)	2191
81	(K J B F G L A I H D E C)	2619
82	(E C D B H J K L A F G I)	2281
83	(L I B H E D C J F K A G)	2189

84 (L D I A G B J K H C E F) 2299
85 (F B K I C E D A H L G J) 2219
86 (D G B K H C A F J E L I) 1978
87 (B D I H F J L K C G E A) 2336
88 (J F D E A B H K C L G I) 2501
89 (J D F H B E A K L G C I) 2366
90 (K A J G F L D B I H C E) 1856
91 (K D I E J H F L G A B C) 2224
92 (B G C D L J F H E A I K) 2588
93 (E C F A H G J D K B L I) 1668
94 (L J F E G H I K B C D A) 2555
95 (B K L C F J D G I A E H) 2672
96 (L D E K C F H A J I G B) 1795
97 (E L B H J G A C I D K F) 1543
98 (D I K J L F E G B C A H) 2061
99 (C H I E K A L D G F B J) 1863
100 (L K F E D J A I H G C B) 2298

Average fitness = 2185.61

Sampling...

the sample of individuals ...
81 (K J B F G L A I H D E C) 2619
30 (G K A F C J E H B L D I) 2403
94 (L J F E G H I K B C D A) 2555
71 (J A C B G I E F D K H L) 2204
1 (K A F G C I H J E L D B) 2194
91 (K D I E J H F L G A B C) 2224
85 (F B K I C E D A H L G J) 2219
25 (J D G F L K A H C B I E) 1805

the most fit of the sample...
25 (J D G F L K A H C B I E) 1805

Sampling...

the sample of individuals ...
40 (B J K A I L H D E F G C) 2434

```
78      (B C K H G A E L D I J F)    2250
18      (A J H F C G I K D L B E)    1787
12      (K D E G A B I F C H J L)    1939
15      (L F J D G H K E I A C B)    2102
26      (A G J C K F D E B L H I)    1995
42      (E F G A K H I C L J D B)    2526
15      (L F J D G H K E I A C B)    2102

the most fit of the sample...
18      (A J H F C G I K D L B E)    1787


Sampling...

the sample of individuals ...
35      (B G I J D E H K F L A C)    2178
74      (F J I C G B A H L K D E)    2149
93      (E C F A H G J D K B L I)    1668
99      (C H I E K A L D G F B J)    1863
31      (F I H K G B D J L E C A)    2249
19      (B I L E G H K C F J D A)    2243
16      (L E I J F K G C B D H A)    2035
17      (I J L F B G H E D A K C)    2311

the most fit of the sample...
93      (E C F A H G J D K B L I)    1668


NIL
CL-USER>
```

Code:

```
( defconstant *population-size* 100 )
( defconstant *selection-size* 8 )
( setf *fitness* #'fitness-distance )

( defclass population ()
```

```
  (
    ( individuals :accessor population-individuals :initarg :individuals )
    ( generation :accessor population-generation :initform 0 )
  )
)

( defmethod size ( ( p population ) )
  ( length ( population-individuals p ) ) )
)

( defmethod display ( ( p population ) )
  ( terpri )( terpri )
  ( princ "Generation " )
  ( prin1 ( population-generation p ) )
  ( princ " Population ..." )
  ( terpri )( terpri )
  ( dolist ( i ( population-individuals p ) )
    ( display i )
  )
  ( terpri )
)

( defmethod initial-population ( &aux individuals )
  ( setf individuals () )
  ( dotimes ( i *population-size* )
    ( push ( new-individual ( + i 1 ) ( tsp-string ) ) individuals )
  )
  ( make-instance 'population :individuals ( reverse individuals ) )
)

( defmethod average ( ( p population ) &aux ( sum 0 ) )
  ( dolist ( i ( population-individuals p ) )
    ( setf sum ( + sum ( funcall *fitness* i ) ) )
  )
  ( setf size ( length ( population-individuals p ) ) )
  ( setf average ( / sum size ) )
  ( float average )
)

( setf *select-demo* nil )
```

```
( defmethod select-individual ( ( p population ) &aux i candidates rn )
  ( setf candidates ( select-individuals p ) )
  ( setf mfi ( most-fit-individual candidates ) )
  ( if *select-demo* ( select-demo-helper candidates mfi ) )
  mfi
)

( defmethod select-individuals ( ( p population ) &aux individuals candidates rn )
  ( setf individuals ( population-individuals p ) )
  ( setf candidates () )
  ( dotimes ( i *selection-size* )
    ( setf rn ( random ( - *population-size* 1 ) ) )
    ( push ( nth rn individuals ) candidates )
  )
  candidates
)

( defmethod most-fit-individual ( ( li list ) &aux max-value max-individual )
  ( setf max-value 10000)
  ( setf max-individual ())
  ( dolist (i li)
    ( if ( < ( funcall *fitness* i ) max-value )
      ( let ()
       ( setf max-value ( funcall *fitness* i ) )
       ( setf max-individual i )
      )
          )
  )
  max-individual
)

( defmethod select-demo-helper ( ( li list )( i individual ) )
  ( princ "the sample of individuals ...") ( terpri )
  ( mapcar #'display li )
  ( terpri )
  ( princ "the most fit of the sample..." )( terpri )
  ( display i )
  ( terpri )
)
```

```
( defmethod population-demo ( &aux p )
  ( setf p ( initial-population ) )
  ( display p)
  ( format t "Average fitness = ~A~%~%" ( average p ) )
  ( setf *select-demo* t )
  ( format t "Sampling...~%~%" )
  ( select-individual p )( terpri )
  ( format t "Sampling...~%~%" )
  ( select-individual p )( terpri )
  ( format t "Sampling...~%~%" )
  ( select-individual p )( terpri )
)
```

Task 8 Mutation pt 2:

Demo:

```
CL-USER> ( setf i ( random-individual ) )
#<INDIVIDUAL {100386BE63}>
CL-USER> ( display i )
0       (D B A J I L E H C G K F)    2048
NIL
CL-USER> ( display ( mutate i ) )
0       (D B A J I L E H C G K F)    2048
NIL
CL-USER> ( display i )
0       (D B A J I L E H C G K F)    2048
NIL
CL-USER> ( display ( mutate i ) )
0       (D B A J I H E L C G K F)    2447
NIL
CL-USER> ( display i )
0       (D B A J I H E L C G K F)    2048
NIL
CL-USER> ( display ( maybe-mutate i ) )
0       (D B A J I H E L C G K F)    2048
NIL
CL-USER> ( display ( maybe-mutate i ) )
0       (D B A J I H E L C K G F)    2620
NIL
CL-USER> ( display ( maybe-mutate i ) )
0       (D B A J I H F L C K G E)    2512
NIL
CL-USER> ( display ( maybe-mutate i ) )
```

```
0      (D B E J I H F L C K G A)   2305
NIL
CL-USER> ( mutate-demo )
0      (G H I L K A E F J B D C)   2276
0      (G A I L K H E F J B D C)   2388
0      (H A I L K G E F J B D C)   2195
0      (D A I L K G E F J B H C)   2228
0      (D A I L K G E F J B H C)   2228
0      (D A G L K I E F J B H C)   2012
0      (D G A L K I E F J B H C)   2025
0      (D G A L K I E B J F H C)   2077
0      (D G A L K E I B J F H C)   1815
0      (D J A L K E I B G F H C)   1870
0      (D J A F K E I B G L H C)   1875
0      (D J A F L E I B G K H C)   2117
0      (D J F A L E I B G K H C)   2267
0      (D H F A L E I B G K J C)   2443
0      (E H F A L D I B G K J C)   2317
0      (E H F A G D I B L K J C)   2011
0      (E H F C G D I B L K J A)   1995
0      (E H F C G D I B L K J A)   1995
0      (E H F G C D I B L K J A)   2057
0      (E H F C G D I B L K J A)   1995
0      (E H F I G D C B L K J A)   2396
NIL
CL-USER> ( maybe-mutate-demo )
0      (K H J I C G B A L F D E)   2234
0      (C H J I K G B A L F D E)   2071   *
0      (C H J I K G B A L F D E)   2071
0      (C H J I K D B A L F G E)   2006   *
0      (C H J I K D B A L F G E)   2006
0      (L H J I K D B A C F G E)   2016   *
0      (L H J I K D B A C F G E)   2016
0      (L K J I H D B A C F G E)   2173   *
0      (L K J I H D B A C F G E)   2173
0      (L K J I H D B A C F G E)   2173
0      (L K J I H D B A C F G E)   2173
0      (L G J I H D B A C F K E)   1739   *
0      (L G J I H D B A C F K E)   1739
0      (L G J I H D B A C F K E)   1739
```

```
0       (L G J I H D B A C F K E)    1739
0       (L G J I H D B A C F K E)    1739
0       (L G F I H D B A C J K E)    2052     *
0       (L G F I H D B A C J K E)    2052
0       (L G F I A D B H C J K E)    1930     *
0       (L G F I A D B H C J K E)    1930
0       (L G F I A D B H C J E K)    1912     *
NIL
```

Code:

```
( defmethod mutate ( ( i individual ) &aux mutation )
  ( setf mutation ( mutation ( individual-tsp-string i ) ) )
  ( make-instance 'individual
    :number ( individual-number i )
    :tsp-string mutation
    :fitness ( funcall *fitness* mutation )
  )
)

( defconstant *pc-m* 50 )

( defmethod maybe-mutate ( ( i individual ) )
  ( if ( <= ( + 1 ( random 100 ) ) *pc-m* )
    ( mutate i )
    i
  )
)

( defmethod mutate-demo ()
  ( setf i ( random-individual ) )
  ( display i )
  ( dotimes ( x 20 )
    ( setf i ( mutate i ) )
    ( display i )
  )


)
( defmethod maybe-mutate-demo ()
  ( setf i ( random-individual ) )
```

```
  ( display i )
  ( dotimes ( x 20 )
    ( setf n ( maybe-mutate i ) )
    ( display-nnl n )
    ( if ( not ( equal n i ) ) ( princ " *" ) )
    ( terpri )
    ( setf i n )
  )
)
```

Task 9 Copy:

Demo:

```
CL-USER> ( perform-copies-demo )

Generation 1 Population ...


the sample of individuals ...
6       (J L B C K G A D F E H I)    2443
59      (D H B J L A K E C I F G)    2563
9       (G A J L I H D K B F E C)    2252
87      (J H K E F A I D B L G C)    1761
30      (E H A I D B K F C J L G)    2054
78      (I B K G H D L C F A E J)    2247
60      (E K B F I C G L J D A H)    1906
68      (L G I J E B C F D K H A)    2105

the most fit of the sample...
87      (J H K E F A I D B L G C)    1761

Selected individual =
87      (J H K E F A I D B L G C)    1761
Possibly muted individual =
87      (J H K E F A I D B L G C)    1761
Renumbered individual =
1       (J H K E F A I D B L G C)    1761


Generation 1 Population ...

1       (J H K E F A I D B L G C)    1761

the sample of individuals ...
40      (B A C I L K J F D H E G)    2517
12      (L H E B G K J F A C D I)    2460
42      (F K J H E B I C D G A L)    2510
43      (L F I C A E H J B D K G)    2314
83      (F A C D H J G B I E K L)    1781
48      (F L B C D I J A G H K E)    1805
65      (B C J H K F E L A G D I)    2307
61      (J G C B K F L A D H E I)    2356
```

the most fit of the sample...
83 (F A C D H J G B I E K L) 1781

Selected individual =
83 (F A C D H J G B I E K L) 1781
Possibly muted individual =
83 (A F C D H J G B I E K L) 1908
Renumbered individual =
2 (A F C D H J G B I E K L) 1908

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908

the sample of individuals ...
87 (J H K E F A I D B L G C) 1761
16 (E G L F C K B H I A D J) 2345
76 (H F G L D E J B K C I A) 2362
91 (C B L G I D E F K A J H) 2113
68 (L G I J E B C F D K H A) 2105
99 (F C E K I D G A B H J L) 2019
37 (F G I J E L H B K C A D) 2292
47 (G J D C K E F A L H I B) 2233

the most fit of the sample...
87 (J H K E F A I D B L G C) 1761

Selected individual =
87 (J H K E F A I D B L G C) 1761
Possibly muted individual =
87 (J H K E F A I D B L G C) 1761
Renumbered individual =
3 (J H K E F A I D B L G C) 1761

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761

the sample of individuals ...
25 (D C L J E H B G K I A F) 2988
59 (D H B J L A K E C I F G) 2563
99 (F C E K I D G A B H J L) 2019
6 (J L B C K G A D F E H I) 2443
77 (C K D F G A J H B L I E) 1978
24 (I A E C J G K L D F H B) 2506
47 (G J D C K E F A L H I B) 2233
64 (J G I F C H B E L D K A) 1805

the most fit of the sample...
64 (J G I F C H B E L D K A) 1805

Selected individual =
64 (J G I F C H B E L D K A) 1805
Possibly muted individual =
64 (J G D F C H B E L I K A) 1807
Renumbered individual =
4 (J G D F C H B E L I K A) 1807

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761
4 (J G D F C H B E L I K A) 1807

the sample of individuals ...
37 (F G I J E L H B K C A D) 2292
49 (I K D G L A B F J E C H) 2465
3 (J H K E F A I D B L G C) 1761
20 (I G E B D C A L H J F K) 2158
62 (K J G L E I H C B D F A) 1960
97 (I B C J L H F D A G K E) 2098
86 (G J L F E K A B H C I D) 1881

40 (B A C I L K J F D H E G) 2517

the most fit of the sample...
3 (J H K E F A I D B L G C) 1761

Selected individual =
3 (J H K E F A I D B L G C) 1761
Possibly muted individual =
3 (J H K E F A I D B L G C) 1761
Renumbered individual =
5 (J H K E F A I D B L G C) 1761

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761
4 (J G D F C H B E L I K A) 1807
5 (J H K E F A I D B L G C) 1761

the sample of individuals ...
19 (B E H K F A G J L I D C) 2110
58 (A E G C L K H J D F I B) 2311
44 (C B L H E I A D J K G F) 2758
14 (E A I F K G J D C B L H) 2462
44 (C B L H E I A D J K G F) 2758
12 (L H E B G K J F A C D I) 2460
70 (E G A J H K F L D I C B) 2284
89 (G A I J C E L H F D K B) 2256

the most fit of the sample...
19 (B E H K F A G J L I D C) 2110

Selected individual =
19 (B E H K F A G J L I D C) 2110
Possibly muted individual =
19 (B E H K F A G J L I D C) 2110
Renumbered individual =
6 (B E H K F A G J L I D C) 2110

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761
4 (J G D F C H B E L I K A) 1807
5 (J H K E F A I D B L G C) 1761
6 (B E H K F A G J L I D C) 2110

the sample of individuals ...
70 (E G A J H K F L D I C B) 2284
23 (J L B E A H G C F K D I) 1623
30 (E H A I D B K F C J L G) 2054
60 (E K B F I C G L J D A H) 1906
9 (G A J L I H D K B F E C) 2252
61 (J G C B K F L A D H E I) 2356
51 (J E F L I C A G D K H B) 2138
5 (G H I F J B E L D C A K) 1942

the most fit of the sample...
23 (J L B E A H G C F K D I) 1623

Selected individual =
23 (J L B E A H G C F K D I) 1623
Possibly muted individual =
23 (J L B E A H G C F K D I) 1623
Renumbered individual =
7 (J L B E A H G C F K D I) 1623

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761
4 (J G D F C H B E L I K A) 1807
5 (J H K E F A I D B L G C) 1761
6 (B E H K F A G J L I D C) 2110

7 (J L B E A H G C F K D I) 1623

the sample of individuals ...
39 (B K D H C A L G E J F I) 2212
38 (E K D L A I G B J H C F) 1894
19 (B E H K F A G J L I D C) 2110
78 (I B K G H D L C F A E J) 2247
15 (G B C J I K F E L D H A) 1980
43 (L F I C A E H J B D K G) 2314
98 (L D I A J F C B E H G K) 2384
24 (I A E C J G K L D F H B) 2506

the most fit of the sample...
38 (E K D L A I G B J H C F) 1894

Selected individual =
38 (E K D L A I G B J H C F) 1894
Possibly muted individual =
38 (E K D L A I G B J H F C) 2004
Renumbered individual =
8 (E K D L A I G B J H F C) 2004

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761
4 (J G D F C H B E L I K A) 1807
5 (J H K E F A I D B L G C) 1761
6 (B E H K F A G J L I D C) 2110
7 (J L B E A H G C F K D I) 1623
8 (E K D L A I G B J H F C) 2004

the sample of individuals ...
64 (J G D F C H B E L I K A) 1805
48 (F L B C D I J A G H K E) 1805
15 (G B C J I K F E L D H A) 1980
20 (I G E B D C A L H J F K) 2158
10 (D C I L K F G A B J H E) 2357

85 (C D A E F J K L G I B H) 2365
31 (B I F D L K J H A E C G) 2047
48 (F L B C D I J A G H K E) 1805

the most fit of the sample...
48 (F L B C D I J A G H K E) 1805

Selected individual =
48 (F L B C D I J A G H K E) 1805
Possibly muted individual =
48 (F K B C D I J A G H L E) 2033
Renumbered individual =
9 (F K B C D I J A G H L E) 2033

Generation 1 Population ...

1 (J H K E F A I D B L G C) 1761
2 (A F C D H J G B I E K L) 1908
3 (J H K E F A I D B L G C) 1761
4 (J G D F C H B E L I K A) 1807
5 (J H K E F A I D B L G C) 1761
6 (B E H K F A G J L I D C) 2110
7 (J L B E A H G C F K D I) 1623
8 (E K D L A I G B J H F C) 2004
9 (F K B C D I J A G H L E) 2033

the sample of individuals ...
56 (G J I H K A L E F D B C) 2265
84 (H G K F L C J I E D A B) 2276
35 (F L I B A E J H D G C K) 2346
21 (F H G J C L D K E A I B) 1985
76 (H F G L D E J B K C I A) 2362
85 (C D A E F J K L G I B H) 2365
36 (A G F I J B L E K C H D) 1662
52 (G I L J F K B H C A E D) 1822

the most fit of the sample...
36 (A G F I J B L E K C H D) 1662

```
Selected individual =
36      (A G F I J B L E K C H D)    1662
Possibly muted individual =
36      (A G F D J B L E K C H I)    1683
Renumbered individual =
10      (A G F D J B L E K C H I)    1683


Generation 1 Population ...

1       (J H K E F A I D B L G C)    1761
2       (A F C D H J G B I E K L)    1908
3       (J H K E F A I D B L G C)    1761
4       (J G D F C H B E L I K A)    1807
5       (J H K E F A I D B L G C)    1761
6       (B E H K F A G J L I D C)    2110
7       (J L B E A H G C F K D I)    1623
8       (E K D L A I G B J H F C)    2004
9       (F K B C D I J A G H L E)    2033
10      (A G F D J B L E K C H I)    1683

NIL
```

Code:

```
( setf *copy-demo* nil )
( defconstant *pc-c* 40 )

( defmethod perform-copies ( ( cp population ) ( np population))
  ( dotimes ( i ( nr-copies ) )
    ( perform-one-copy cp np )
  )
)

( defmethod nr-copies ()
  ( * ( / *pc-c* 100 ) *population-size* )
)
```

```
( defmethod perform-one-copy ( ( cp population ) ( np population ) &aux x m mm new-i )
  ( setf m ( select-individual cp ) )
  ( if *copy-demo* ( format t "Selected individual = ~%" ) )
  ( if *copy-demo* ( display m ) )
  ( setf mm ( maybe-mutate m ) )
  ( if *copy-demo* ( format t "Possibly muted individual = ~&" ) )
  ( if *copy-demo* ( display mm ) )
  ( setf ( individual-number mm ) ( + 1 ( size np ) ) )
  ( if *copy-demo* ( format t "Renumbered individual = ~&" ) )
  ( if *copy-demo* ( display mm ) )
  ( setf new-i ( new-individual ( + 1 ( size np ) ) ( individual-tsp-string mm ) ) )
  ( setf
    ( population-individuals np )
    ( append ( population-individuals np ) ( list new-i ) )
  )
  nil
)

( defmethod empty-population ( ( cp population ) &aux np )
  ( setf np ( make-instance 'population ) )
  ( setf ( population-individuals np ) () )
  ( setf ( population-generation np ) ( + 1 ( population-generation cp ) ) )
  np
)

( defmethod perform-copies-demo ( &aux cp np )
  ( setf cp ( initial-population ) )
  ( setf np ( empty-population cp ) )
  ( display np )
  ( setf *select-demo* t )
  ( setf *copy-demo* t )
  ( dotimes ( i 10 )
    ( perform-one-copy cp np )
    ( display np )
  )
  ( setf *select-demo* nil )
  ( setf *copy-demo* nil )
  nil
)
```

Task 10 Crossover pt 2:

Demo:

```
CL-USER> ( perform-crossovers-demo )


Generation 1 Population ...


the sample of individuals ...
61      (C D H A F B I L E G K J)    2091
77      (D C H B F A E I J G K L)    2226
36      (G H B A E C J I L F D K)    2223
3       (B E A C J F D K L G I H)    2145
10      (A F J C D K L H G B E I)    2215
61      (C D H A F B I L E G K J)    2091
37      (E I K F A B G H C D J L)    2087
27      (L J A I K F B G C E D H)    2460

the most fit of the sample...
37      (E I K F A B G H C D J L)    2087

the sample of individuals ...
45      (E J L H A K C F B D I G)    2146
53      (G D A J K B C L F E I H)    2577
69      (H F G I D C J L E K B A)    1966
64      (B J F D A C E I K L H G)    2177
82      (I H C D J E G L F K B A)    2338
4       (E A B D G C J H I F L K)    2186
80      (C K L B E J I F G A D H)    2120
33      (J I G D A K E B L F H C)    1629

the most fit of the sample...
33      (J I G D A K E B L F H C)    1629

Select mother =
37      (E I K F A B G H C D J L)    2087
Select father =
33      (J I G D A K E B L F H C)    1629
The crossover =
0       (E I K F A B G H C D J L)    2087
```

The possibly mutated individual =
0 (E I K F A B G H C D J L) 2087
The renumberd individual =
0 (E I K F A B G H C D J L) 2087

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087

the sample of individuals ...
99 (G J C A D I E L K F B H) 1690
54 (L I H B G E F D A C J K) 2281
11 (K F I E A D L G B C J H) 2324
85 (K C I F E B D J L A G H) 2136
30 (I L E B J C K D H G A F) 2028
54 (L I H B G E F D A C J K) 2281
91 (H G J D C B I A E L F K) 2154
52 (H B A K F E I L J C G D) 2367

the most fit of the sample...
99 (G J C A D I E L K F B H) 1690

the sample of individuals ...
24 (D H F G J A I B K L E C) 2246
34 (G H C F J L A I D B K E) 1770
67 (F J A K B C E H D G I L) 2741
12 (A G D I B C J F L K H E) 2257
34 (G H C F J L A I D B K E) 1770
74 (F H A L G D K I J B C E) 2227
49 (E J C D A I F G B H L K) 2410
51 (H G A I C B D J L K F E) 2164

the most fit of the sample...
34 (G H C F J L A I D B K E) 1770

Select mother =
99 (G J C A D I E L K F B H) 1690
Select father =
34 (G H C F J L A I D B K E) 1770

The crossover =
0 (G J C A I D E L K F B H) 1697
The possibly mutated individual =
0 (G H C A I D E L K F B J) 1544
The renumberd individual =
1 (G H C A I D E L K F B J) 1544

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544

the sample of individuals ...
34 (G H C F J L A I D B K E) 1770
47 (H K D A F B G C J E L I) 2217
54 (L I H B G E F D A C J K) 2281
51 (H G A I C B D J L K F E) 2164
15 (G F D K H L C A E B J I) 2162
7 (K G L C D I E J F H B A) 2626
35 (A D E G I B C J K L F H) 2307
51 (H G A I C B D J L K F E) 2164

the most fit of the sample...
34 (G H C F J L A I D B K E) 1770

the sample of individuals ...
70 (G K L B H D F J C I E A) 2336
89 (E B L I H F J G K D C A) 1844
21 (F E I J B K L G A D H C) 1989
84 (B J C E G F K A I L H D) 2670
17 (J I D E G K B F C L H A) 2008
26 (E G I A L F D C K J H B) 2699
34 (G H C F J L A I D B K E) 1770
98 (B D G L I F J E C K H A) 2312

the most fit of the sample...
34 (G H C F J L A I D B K E) 1770

Select mother =

34 (G H C F J L A I D B K E) 1770
Select father =
34 (G H C F J L A I D B K E) 1770
The crossover =
0 (G H C F J L A I D B K E) 1770
The possibly mutated individual =
0 (G H C F J L A I D B K E) 1770
The renumberd individual =
2 (G H C F J L A I D B K E) 1770

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770

the sample of individuals ...
93 (B D K L A C I E J F H G) 2134
12 (A G D I B C J F L K H E) 2257
2 (J D I C A L B E H K G F) 2235
35 (A D E G I B C J K L F H) 2307
45 (E J L H A K C F B D I G) 2146
17 (J I D E G K B F C L H A) 2008
63 (H D F J L A C E G I B K) 2178
40 (F C A E D B H L G I K J) 2042

the most fit of the sample...
17 (J I D E G K B F C L H A) 2008

the sample of individuals ...
37 (E I K F A B G H C D J L) 2087
7 (K G L C D I E J F H B A) 2626
28 (J K E G B A I D F C L H) 2350
10 (A F J C D K L H G B E I) 2215
56 (L B J G I D F H E A K C) 2292
90 (A E B C K J D F G I L H) 2445
96 (F E I C J H L G K B A D) 2680
54 (L I H B G E F D A C J K) 2281

the most fit of the sample...
37 (E I K F A B G H C D J L) 2087

Select mother =
17 (J I D E G K B F C L H A) 2008
Select father =
37 (E I K F A B G H C D J L) 2087
The crossover =
0 (J I D E G K B F C L H A) 2008
The possibly mutated individual =
0 (J I D E G K B F C A H L) 1782
The renumberd individual =
3 (J I D E G K B F C A H L) 1782

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770
4 (J I D E G K B F C A H L) 1782

the sample of individuals ...
56 (L B J G I D F H E A K C) 2292
54 (L I H B G E F D A C J K) 2281
50 (C I K D A F B H J E G L) 2394
54 (L I H B G E F D A C J K) 2281
67 (F J A K B C E H D G I L) 2741
79 (C H I J L K F E D B G A) 1752
6 (F A C K E I D B G L J H) 1797
60 (L A H K E D I J G B C F) 1899

the most fit of the sample...
79 (C H I J L K F E D B G A) 1752

the sample of individuals ...
4 (E A B D G C J H I F L K) 2186
57 (H L E B I D G A J C K F) 2078
28 (J K E G B A I D F C L H) 2350
9 (D C B G K I A E F H J L) 2745

67 (F J A K B C E H D G I L) 2741
1 (E H F L G C K J A B I D) 2384
29 (B I L K F J C D H A E G) 2239
20 (K I J B G H A F L E D C) 1758

the most fit of the sample...
20 (K I J B G H A F L E D C) 1758

Select mother =
79 (C H I J L K F E D B G A) 1752
Select father =
20 (K I J B G H A F L E D C) 1758
The crossover =
0 (C H I J L K F E D B G A) 1752
The possibly mutated individual =
0 (C H I K L J F E D B G A) 1995
The renumberd individual =
4 (C H I K L J F E D B G A) 1995

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770
4 (J I D E G K B F C A H L) 1782
5 (C H I K L J F E D B G A) 1995

the sample of individuals ...
64 (B J F D A C E I K L H G) 2177
98 (B D G L I F J E C K H A) 2312
55 (J G C H E B K L D F A I) 1822
10 (A F J C D K L H G B E I) 2215
35 (A D E G I B C J K L F H) 2307
23 (J E C H G F I K D B L A) 1897
5 (J K C D L F G H B A I E) 2400
63 (H D F J L A C E G I B K) 2178

the most fit of the sample...
55 (J G C H E B K L D F A I) 1822

the sample of individuals ...
86 (F I E A J H G K D L B C) 2178
78 (B D C F K E G J I L A H) 1692
85 (K C I F E B D J L A G H) 2136
7 (K G L C D I E J F H B A) 2626
18 (A L F I K G D H B E C J) 2505
13 (D J K B L G I F H A C E) 1915
70 (G K L B H D F J C I E A) 2336
99 (G J C A D I E L K F B H) 1690

the most fit of the sample...
99 (G J C A D I E L K F B H) 1690

Select mother =
55 (J G C H E B K L D F A I) 1822
Select father =
99 (G J C A D I E L K F B H) 1690
The crossover =
0 (G J C H E B K L D F A I) 1982
The possibly mutated individual =
0 (G J C H E B K L D F A I) 1982
The renumberd individual =
5 (G J C H E B K L D F A I) 1982

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770
4 (J I D E G K B F C A H L) 1782
5 (C H I K L J F E D B G A) 1995
6 (G J C H E B K L D F A I) 1982

the sample of individuals ...
4 (E A B D G C J H I F L K) 2186
72 (I H B J F L G A D C E K) 2342
46 (E F B G C K A L J I D H) 2297
86 (F I E A J H G K D L B C) 2178

22 (B G L H D J C K E A I F) 2574
71 (E L F J D A H I K C B G) 2353
47 (H K D A F B G C J E L I) 2217
34 (G H C F J L A I D B K E) 1770

the most fit of the sample...
34 (G H C F J L A I D B K E) 1770

the sample of individuals ...
70 (G K L B H D F J C I E A) 2336
70 (G K L B H D F J C I E A) 2336
20 (K I J B G H A F L E D C) 1758
58 (L J B A F I K H G C D E) 2259
19 (H C L E D F A J G I B K) 1743
3 (B E A C J F D K L G I H) 2145
71 (E L F J D A H I K C B G) 2353
58 (L J B A F I K H G C D E) 2259

the most fit of the sample...
19 (H C L E D F A J G I B K) 1743

Select mother =
34 (G H C F J L A I D B K E) 1770
Select father =
19 (H C L E D F A J G I B K) 1743
The crossover =
0 (G H C F J L A I D B K E) 1770
The possibly mutated individual =
0 (G H C F J L A I D B K E) 1770
The renumberd individual =
6 (G H C F J L A I D B K E) 1770

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770
4 (J I D E G K B F C A H L) 1782
5 (C H I K L J F E D B G A) 1995

6 (G J C H E B K L D F A I) 1982
7 (G H C F J L A I D B K E) 1770

the sample of individuals ...
68 (B I L A F D E C J G K H) 2240
91 (H G J D C B I A E L F K) 2154
1 (E H F L G C K J A B I D) 2384
30 (I L E B J C K D H G A F) 2028
92 (G F L A I B D E H J C K) 2327
51 (H G A I C B D J L K F E) 2164
94 (I C L K B H A E F G J D) 2152
5 (J K C D L F G H B A I E) 2400

the most fit of the sample...
30 (I L E B J C K D H G A F) 2028

the sample of individuals ...
80 (C K L B E J I F G A D H) 2120
71 (E L F J D A H I K C B G) 2353
92 (G F L A I B D E H J C K) 2327
35 (A D E G I B C J K L F H) 2307
34 (G H C F J L A I D B K E) 1770
8 (H L A G C K J D I B F E) 2191
57 (H L E B I D G A J C K F) 2078
46 (E F B G C K A L J I D H) 2297

the most fit of the sample...
34 (G H C F J L A I D B K E) 1770

Select mother =
30 (I L E B J C K D H G A F) 2028
Select father =
34 (G H C F J L A I D B K E) 1770
The crossover =
0 (I L E B J C K D H G A F) 2028
The possibly mutated individual =
0 (I L D B J C K E H G A F) 1832
The renumberd individual =
7 (I L D B J C K E H G A F) 1832

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770
4 (J I D E G K B F C A H L) 1782
5 (C H I K L J F E D B G A) 1995
6 (G J C H E B K L D F A I) 1982
7 (G H C F J L A I D B K E) 1770
8 (I L D B J C K E H G A F) 1832

the sample of individuals ...
99 (G J C A D I E L K F B H) 1690
76 (L F C I J B G H D K E A) 2056
1 (E H F L G C K J A B I D) 2384
2 (J D I C A L B E H K G F) 2235
2 (J D I C A L B E H K G F) 2235
26 (E G I A L F D C K J H B) 2699
83 (B H E A D K G I L F C J) 2329
1 (E H F L G C K J A B I D) 2384

the most fit of the sample...
99 (G J C A D I E L K F B H) 1690

the sample of individuals ...
74 (F H A L G D K I J B C E) 2227
22 (B G L H D J C K E A I F) 2574
85 (K C I F E B D J L A G H) 2136
82 (I H C D J E G L F K B A) 2338
50 (C I K D A F B H J E G L) 2394
95 (L K B E F A C H I J D G) 1760
50 (C I K D A F B H J E G L) 2394
18 (A L F I K G D H B E C J) 2505

the most fit of the sample...
95 (L K B E F A C H I J D G) 1760

Select mother =
99 (G J C A D I E L K F B H) 1690

Select father =
95 (L K B E F A C H I J D G) 1760
The crossover =
0 (G J C A D I L K B E F H) 1727
The possibly mutated individual =
0 (G J C A D I L K B E F H) 1727
The renumberd individual =
8 (G J C A D I L K B E F H) 1727

Generation 1 Population ...

1 (E I K F A B G H C D J L) 2087
2 (G H C A I D E L K F B J) 1544
3 (G H C F J L A I D B K E) 1770
4 (J I D E G K B F C A H L) 1782
5 (C H I K L J F E D B G A) 1995
6 (G J C H E B K L D F A I) 1982
7 (G H C F J L A I D B K E) 1770
8 (I L D B J C K E H G A F) 1832
9 (G J C A D I L K B E F H) 1727

the sample of individuals ...
93 (B D K L A C I E J F H G) 2134
38 (L H I E K B F J A G C D) 2188
54 (L I H B G E F D A C J K) 2281
40 (F C A E D B H L G I K J) 2042
59 (C D B F K J L E G H A I) 2094
60 (L A H K E D I J G B C F) 1899
49 (E J C D A I F G B H L K) 2410
78 (B D C F K E G J I L A H) 1692

the most fit of the sample...
78 (B D C F K E G J I L A H) 1692

the sample of individuals ...
6 (F A C K E I D B G L J H) 1797
37 (E I K F A B G H C D J L) 2087
63 (H D F J L A C E G I B K) 2178
53 (G D A J K B C L F E I H) 2577

```
82      (I H C D J E G L F K B A)    2338
64      (B J F D A C E I K L H G)    2177
39      (J E A D B C H L F G I K)    2258
75      (B E A G K L J F D I C H)    2219

the most fit of the sample...
6       (F A C K E I D B G L J H)    1797

Select mother =
78      (B D C F K E G J I L A H)    1692
Select father =
6       (F A C K E I D B G L J H)    1797
The crossover =
0       (B D C F K E G J I L A H)    1692
The possibly mutated individual =
0       (B D C F K E G J I L A H)    1692
The renumberd individual =
9       (B D C F K E G J I L A H)    1692


Generation 1 Population ...

1       (E I K F A B G H C D J L)    2087
2       (G H C A I D E L K F B J)    1544
3       (G H C F J L A I D B K E)    1770
4       (J I D E G K B F C A H L)    1782
5       (C H I K L J F E D B G A)    1995
6       (G J C H E B K L D F A I)    1982
7       (G H C F J L A I D B K E)    1770
8       (I L D B J C K E H G A F)    1832
9       (G J C A D I L K B E F H)    1727
10      (B D C F K E G J I L A H)    1692

NIL
```

Code:

```
( setf *crossover-demo* nil )
```

```
( defconstant *pc-x* 60 )

( defmethod perform-crossovers ( ( cp population ) ( np population ) )
  ( dotimes ( i ( nr-crossovers ) )
    ( perform-one-crossover cp np )
  )
)

( defmethod nr-crossovers ()
  ( * ( / *pc-x* 100 ) *population-size* )
)

( defmethod perform-one-crossover ( ( cp population ) ( np population ) )
  ( let ( x m mm mother father new-i )
    ( setf mother ( select-individual cp ) )
    ( setf father ( select-individual cp ) )
    ( if *crossover-demo* ( format t "Select mother = ~%" ) )
    ( if *crossover-demo* ( display mother ) )
    ( if *crossover-demo* ( format t "Select father = ~&" ) )
    ( if *crossover-demo* ( display father ) )
    ( setf m ( crossover mother father ) )
    ( if *crossover-demo* ( format t "The crossover = ~&" ) )
    ( if *crossover-demo* ( display m ) )
    ( setf mm ( maybe-mutate m ) )
    ( if *crossover-demo* ( format t "The possibly mutated individual = ~&" ) )
    ( if *crossover-demo* ( display mm ) )
    ( setf ( individual-number mm ) ( + ( size np ) ) )
    ( if *crossover-demo* ( format t "The renumberd individual = ~&" ) )
    ( if *crossover-demo* ( display mm ) )
    ( setf new-i ( new-individual ( + 1 ( size np ) ) ( individual-tsp-string mm ) ) )
    ( setf ( population-individuals np )
            ( append ( population-individuals np ) ( list new-i ) )
          )
  )
  nil
)

( defmethod crossover ( ( mother individual ) ( father individual ) &aux mi fi x i )
  ( setf mi (individual-tsp-string mother ) )
  ( setf fi (individual-tsp-string father ) )
  ( setf x ( crossover mi fi ) )
  ( setf i ( new-individual 0 x ) )
  ( if ( eq x nil )
    ( setf i( new-individual 0 mi ) )
```

```
        ( setf i ( new-individual 0 x ) )
  )
  i
)


( defmethod perform-crossovers-demo ( &aux cp np )
  ( setf cp ( initial-population ) )
  ( setf np ( empty-population cp ) )
  ( display np )
  ( setf *select-demo* t )
  ( setf *crossover-demo* t )
  ( dotimes ( i 10 )
    ( perform-one-crossover cp np )
    ( display np )
  )
  ( setf *select-demo* nil )
  ( setf *crossover-demo* nil )
  nil
)
```

Task 11 The GA:

Demo:

```
CL-USER> ( ga )
```

Generation 0 Population ...

1	(F J I B G D C L H K E A)	2343
2	(L D K B E A I G C F H J)	2097
3	(J L D A C B K G F E H I)	2256
4	(I A B C H G F J K L D E)	2200
5	(L B C E F I G K J D A H)	2397
6	(E L G A C I D J F H K B)	2064
7	(J I A L F H K B D C E G)	2449
8	(F B D L J I A C E G K H)	2252
9	(E K J I G B C D H A F L)	2057
10	(K I B F A G J C H D L E)	1651
11	(F G H L B E C I A J D K)	2335
12	(K E H D L B F C J I A G)	1912
13	(B H I G K F L E D C J A)	2210
14	(L C A H F I E D B J K G)	1997
15	(A F C E D I B H G J L K)	1691
16	(F E L C G H A D K I B J)	1821
17	(G I B F C J H L A E K D)	2177
18	(J L I B E G H D K C F A)	1954
19	(F A K B H E J I L D C G)	2128
20	(C B H J L K D G A I E F)	2187
21	(L G E I H F A K C J D B)	2501
22	(B C D K J I G E L H F A)	2430
23	(J G K B H D C F A L I E)	2074
24	(I F C G J A H L D B E K)	1519
25	(G A K F D I H B L J E C)	2343
26	(B K D G L F I E C J A H)	2142
27	(E F D J K I C B G A H L)	2346
28	(I J D B K C F A G E L H)	2113
29	(H K J I E D L B A G F C)	2031
30	(A F K D E H C G L J I B)	1876
31	(K F B G I J C A H E L D)	1737
32	(G I A F C H J K L D B E)	1927
33	(B A C E K D H F G L J I)	2052
34	(H D C B K L A G J I F E)	2173
35	(D F G K H J C A I B L E)	1964
36	(D H B G E J C K F L I A)	2610

37	(G J H F E K I L B A C D)	1704
38	(L H I A E J K D B C F G)	2750
39	(G E L A J K C I D F H B)	2626
40	(K I A H B L G D E C J F)	2287
41	(B G F E I C A L H K J D)	2546
42	(J D E A B I G K H L F C)	2420
43	(B F H A I C E J D L K G)	2439
44	(A L F K G J I C B E D H)	2287
45	(F J I B H G L K D E C A)	1716
46	(H J I E L B C D F G K A)	2153
47	(C L I H G A D K F E B J)	1980
48	(K I H J F B D L G E A C)	2075
49	(L D A H K C E F J B G I)	2414
50	(E D I H F C L J A K G B)	2477
51	(H I E K G A J B D C F L)	2062
52	(K B A L F J I C D G H E)	2435
53	(D K J A E B C I H G F L)	2569
54	(H K C E F I G A J B L D)	2250
55	(K D B C I H E G A J F L)	2562
56	(F H C K E I J B G D L A)	1838
57	(J C E B D A L H I K G F)	2548
58	(E F C L J K H G D A B I)	2448
59	(E L C B F D H A J G I K)	2052
60	(B I H J A G L E C D K F)	2271
61	(D F A G J L B I C H K E)	1612
62	(F D G H I J A K L E C B)	2338
63	(A D E J I L C H F K G B)	2130
64	(B D E C G L H F A J I K)	2133
65	(C G J E D B L K F H A I)	1764
66	(I A H K E D C B G F L J)	2133
67	(L B F H E K I A G D J C)	2102
68	(K I E A F J B D L C H G)	1981
69	(B H E I F C K A D L J G)	2349
70	(F L D J A C K H B E I G)	2172
71	(D B F C K H A E J L G I)	2353
72	(K F D J H A L B G I C E)	2085
73	(A C F D B J L G K E H I)	1920
74	(K G C I F D E H B L J A)	2333
75	(I E B G F C K L A J H D)	2426
76	(A K G F D B L J H E I C)	2332

77 (H A F C G L E D K I J B) 1776
78 (L G E D H I K C F A J B) 2527
79 (L G I J B K C E D H A F) 2203
80 (G H F J D E I L B K C A) 1754
81 (H F E C D B A L K I J G) 2278
82 (F G B I D H C J A E K L) 1950
83 (C K F H I B D G J L A E) 2197
84 (E D G B A C J F K L I H) 2030
85 (F I C E L H K A J D G B) 2743
86 (E J K G D F A B L C I H) 2590
87 (B C A G E K J D L F I H) 1972
88 (H F J B G A K E L D C I) 2122
89 (D E I F C J B H G L K A) 2018
90 (L E F D J K C G B I H A) 2061
91 (F C D E K A I B H J G L) 1979
92 (I B H G L D K A J E C F) 2136
93 (C L J B G K F E H I D A) 2548
94 (F C L B E D J H G A K I) 2042
95 (I G A B C K J L E F H D) 2575
96 (L J A I K H D B G F C E) 2578
97 (J L D G K I C F A E B H) 2175
98 (H I G E L F B C K J D A) 2553
99 (G E L D A J I F K C H B) 2149
100 (F G B D E C L H K A J I) 2522

average fitness of population 0 = 2185.66
average fitness of population 1 = 1869.89
average fitness of population 2 = 1780.34
average fitness of population 3 = 1849.31
average fitness of population 4 = 1796.49
average fitness of population 5 = 1763.22
average fitness of population 6 = 1710.78
average fitness of population 7 = 1597.47
average fitness of population 8 = 1535.82
average fitness of population 9 = 1507.4
average fitness of population 10 = 1483.2
average fitness of population 11 = 1458.36
average fitness of population 12 = 1461.71
average fitness of population 13 = 1483.09
average fitness of population 14 = 1443.96

average fitness of population 15 = 1456.95
average fitness of population 16 = 1423.72
average fitness of population 17 = 1462.92
average fitness of population 18 = 1405.12
average fitness of population 19 = 1440.38
average fitness of population 20 = 1374.22
average fitness of population 21 = 1372.71
average fitness of population 22 = 1385.56
average fitness of population 23 = 1343.02
average fitness of population 24 = 1327.06
average fitness of population 25 = 1316.91

Generation 25 Population ...

1	(H A C F K E L B D G J I)	1175
2	(E H C F K A D B L I G J)	1802
3	(H A C F K E D L B I J G)	1072
4	(H C A F K E D B I L J G)	1090
5	(H C A F K E D G L I J B)	1403
6	(A K C F H E D B L I G J)	1125
7	(A H C F K E D B L I J G)	1112
8	(H A C F K E D B L I J G)	1089
9	(L A C F K E D B H I G J)	1563
10	(A K C F H E D B L I G J)	1975
11	(H C A G K E D B L I J F)	1330
12	(A H C F K E I B L D J G)	1230
13	(H A C F K E D L B I J G)	1072
14	(A H C F K E D B L I G J)	1125
15	(A H C F K E D I L B J G)	1198
16	(H C A F K E D B I L J G)	1247
17	(H C A F K E D B L J G I)	1311
18	(H A C F K E D L B I J G)	1072
19	(A H C F K E D B L I G J)	1112
20	(H A C F K E D L B I J G)	1072
21	(A H C F K E D B J I L G)	1318
22	(H A C F K E D L B I J G)	1072
23	(H A C F K E D L B I J G)	1072
24	(H C A F K E I L B D J G)	1225

25	(A H C F K E D B L I G J)	1125
26	(H C A F K E D B L I J G)	1090
27	(G C A F D E K L B I J H)	1280
28	(A H J F K E D B L I G C)	1412
29	(A H C F K E D B L I J G)	1112
30	(H A C E K F L B D I G J)	1257
31	(L C A F K E D B H I G J)	1103
32	(L C A F K E D B H I G J)	1563
33	(A H C F K E D B L I G J)	1125
34	(H C A F K E I B L D J G)	1208
35	(H A C F K J D B L I E G)	1679
36	(A H C F K E D B L I G J)	1125
37	(H C A F K E I B L D J G)	1208
38	(F A C H K E D L B I J G)	1072
39	(A H C F K E D G B I J L)	1437
40	(F A C H K E D L B I J G)	1238
41	(E C A F K H D B L I G J)	1754
42	(H A C F K E D B L I J G)	1089
43	(H A C F K E D B L I J G)	1089
44	(A B C F K E D H L I G J)	1988
45	(J A C F K E D B L I G H)	1276
46	(H A C F I E D B L K J G)	1554
47	(H A C F K E D B G L J I)	1465
48	(D C A F K E I B L H J G)	1626
49	(H A D F K E C L B I J G)	1633
50	(H A C F K E D L B I J G)	1072
51	(H A C F K E D B L I G J)	1102
52	(H A C F K E D B L I J G)	1089
53	(H C A F K E D B L I J G)	1090
54	(I H C F K E D B L A G J)	1506
55	(H A C F K E D L B I J G)	1072
56	(H A C F L E I B K D J G)	1525
57	(H A C F K E I B L D J G)	1207
58	(H C A F K E D B L I G J)	1103
59	(H C A F K E D L B I J G)	1073
60	(H A C F K E D G L I B J)	1300
61	(H A C F K I L B D G J E)	1467
62	(H A C F K E D B L I G J)	1102
63	(H A C F E K D B L I G J)	1150
64	(H C A F K E D L B I J G)	1073

65 (H A C F K E D B L I G J) 1102
66 (H E C F K A D B L I J G) 2088
67 (A H C F J E D B L I K G) 1758
68 (A H C F K E D B L I J G) 1112
69 (H A C F K E D B L I G J) 1102
70 (A H C F K G D B L I J E) 1562
71 (A H B F K E D L C I J G) 1744
72 (H A C E K F D B L J G I) 1428
73 (A H C F K E D B L I G J) 1125
74 (H A G F K E D L B I J C) 1294
75 (H A C F K E D L B I J G) 1072
76 (A H C F K E D B L J I G) 1330
77 (H C A F K E D B L I G J) 1103
78 (H A C F K E D I L B J G) 1175
79 (H A C F K E D L B I J G) 1072
80 (J A C F K E D I L B H G) 1207
81 (A H C F K E D B L I J G) 1112
82 (D A C F K E H B L I G J) 1726
83 (H A C G K E D I L B J F) 1405
84 (H C A F K E D B L I G J) 1103
85 (H A E F K C D L B G J I) 1907
86 (H C A F K E I J L D G B) 1453
87 (H A C F K E D B L I G J) 1102
88 (H D C F K E A B I L G J) 1970
89 (H G C F K E D B L I A J) 1515
90 (H A C F K E D B L I G J) 1102
91 (A H C F K E J B L I D G) 1410
92 (A H C F K E D L B I J G) 1095
93 (H C A F K E D B L I G J) 1103
94 (H A C F K E D L B I J G) 1072
95 (H A C F K E D B L I G J) 1102
96 (H A C F J E D B L I G K) 1673
97 (A H K F C E D L B I J G) 1659
98 (H A C F K E D B L I J G) 1089
99 (A H C F K E J B I L G D) 1549
100 (A H C F K E D G L I B J) 1323

average fitness of population 25 = 1316.91
1316.91

Code:

```
( defconstant *nr-generations* 25 )

( defmethod next-generation ( ( cp population ) &aux np )
  ( setf np ( empty-population cp ) )
  ( perform-copies cp np )
  ( perform-crossovers cp np )
  np
)

( defmethod ga ( &aux p )
  ( setf p ( initial-population ) )
  ( terpri )
  ( summarize p )
  ( dotimes ( i *nr-generations* )
    ( setf p ( next-generation p ) )
    ( check-average p )
  )
  ( terpri )
  ( summarize p )
)

( defmethod summarize ( ( p population ) )
  ( display p )
  ( check-average p )
  ( average p )
)

( defmethod check-average ( ( p population ) )
  ( format t "average fitness of population ~A = ~A~%" ( population-generation p ) (
average p ) )
)
```

Behavior of the GA

This traveling salesman problem genetic algorithm attempts to find the shortest distance tour between cities throughout New York State. Using the evolutionary approach, it creates 25 generations of populations and finds the shortest distance tour between them. The mutations happen by swapping the placement of 2 cities in a tour. The crossovers find the sequence of cities that are the same between the mother and the father tours. It then combines them into a child tour and that is how the child strings are created. I think something I would have liked to do differently would be the crossover. I would probably come up with another way to create the crossover. I think it would be interesting to see what happens if you crossover at the two cities that are farthest apart.