→ Implement Simulated Annealing to solve N-Queue problem

Algorithm

Intilize temperature t.
while the temperature is not zero do
/* Random moves in the state space */

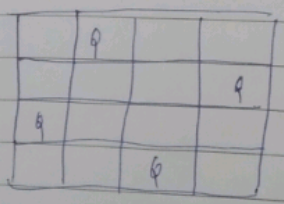
for a predefined number of iterations do
L_{new} = Generate_successor ($L_{current}$)

if $F(t / L_{actual}) - f(L_{new}), T)) > 0$ then $L_{actual} = L_{new}$

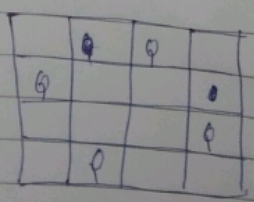
efor
the temperature is decreased.
ewhile.

Solution :-

Solution 1 = [2, 0, 3, 1]



Solution 2 = [1, 3, 0, 2]



Algorithm:

$s \leftarrow s_0$
$e \rightarrow E(s)$
$k \leftarrow 0$
$T_0 = \text{temperature}(s, E)$
while $k < k_{max}$ and $e > e_{max}$ do.

$s_n \leftarrow \text{neighbour}(s)$
$e_n \leftarrow E(s_n)$
if $P(e, e_n, \text{temp}(k)) > \text{random}()$ then
$s \leftarrow s_n$
$e \leftarrow e_n$
end if.
$k \leftarrow k+1$
end while
return s

current ← initial state.
current cost ← cost (current)
$T \leftarrow$ a large positive value.
while $T > 0$ and current - cost > 0.
neighbour ← generated neighbour of current state
neighbour_cost ← cost (neighbour)
if cost_diff > 0
current ← neighbour.
current_cost ← neighbour_cost.
$T = T - 1$
end while
return current cost, current state.