PAGE NO :
DATE :

A* for 8 queens

→ Algorithm

```
func heuristic (state) :
        h=0
    for i in range (len(state)):
        for j in range (i+1, len (state)) :
            if abs(state[i] - state[j]) == abs(i - j) or
                    state[i] == state [j] :
                h+=1
    return h
```

temp

```
func a_star() :
    initial_state = []
    h = [] , g=8
    heap.push(h, ~~initi~~ (heuristic(initial)) + g, initial_state)
    while h :
        c = heap.pop()
        if h(c[1]) + c[2] == 0 :
            return c
        if len(c[1]) == 8 : continue
        for i in range( 1, 9):
            n=c[1] + [i]
            heap.push( h, (heu(n) + c[2]+1),
                                    n, g
```

→ Hill climbing

Algorithm

```
func h(state):
  h=0
  for i in range(len(state)):
      for j in range(i+1, len(state)):
          if abs(state[i] - state[j]) == abs
                or state[i] == state[j]
                h += 1
      return h

func get_neigh(state):
      best = state
    for i in len(state):
      for j in range(~~len~~ 8):
      new = ~~state[i]~~ = state[0:i] + j + state[
          if h(new) < h(best):
                best = new
    return best

func hill_climb():
    ini = [random.randint(1,8) for i in range
    cur = ini
    while True ~~h(cur)>0~~ :                c_h = ~~get~~ h(ini)
        ~~cur = get_neigh(cur)~~
        ~~if h(cur) == 0:~~
            ~~return cur.~~
        ~~c_h = h(cur)~~
        c_h = h(cur)
        cur = get_neigh(cur)
        if h(cur) == 0:
            ret
```

[illegible] 29/10