Steve Tanimoto's latest A*

```
1. For the start state s_0, compute f(s_0) = g(s_0)+h(s_0) = h(s_0) and put [s_0, f(s_0)] on a list (priority queue) OPEN.
```

2. If OPEN is empty, output "DONE" and stop.

3. Find and remove the item [s,p] on OPEN having **lowest** p. Break ties arbitrarily Put [s,p] on CLOSED.

If s is a goal state: output its description (and backtrace a path), and

if h is known to be admissible, halt.

4. Generate the list L of [s',f(s')] pairs where the s' are the successors

of s and their f values are computed using f(s') = g(s')+h(s').

Consider each [s',f(s')].

If there is already a pair [s', q] on CLOSED (for any value q):

if f(s') > q, then remove [s', f(s')] from L.

If $f(s') \le q$, then remove [s',q] from CLOSED.

Else if there is already a pair [s', q] on OPEN (for any value q):

if f(s') > q, then remove [s', f(s')] from L.

If $f(s') \le q$, then remove [s',q] from OPEN.

5. Insert all members of L onto OPEN.

6. Go to Step 2.