New Coverage Algorithm

```
options(digits = 4)
```

```
V = 9
S = (1:V)
alpha = 0.6
END = 9
```

```
# Environment:
#  7 - 8 - 9
#  4 - 5 - 6
#  1 - 2 - 3
R = rbind(c(-1, 0, -1, 0, -1, -1, -1, -1, -1),
          c(0, -1, 0, -1, 0, -1, -1, -1, -1),
          c(-1, 0, -1, -1, -1, 0, -1, -1, -1),
          c(0, -1, -1, -1, 0, -1, 0, -1, -1),
          c(-1, 0, -1, 0, -1, 0, -1, 0, -1),
          c(-1, -1, 0, -1, 0, -1, -1, -1, 100),
          c(-1, -1, -1, 0, -1, -1, -1, 0, -1),
          c(-1, -1, -1, -1, 0, -1, 0, -1, 100),
          c(-1, -1, -1, -1, -1, 0, -1, 0, -1))
Q = matrix(0, 9, 9)
```

```
rounds = 1000
r = 0

get_actions <- function(s) {
  a = c()
  for (i in 1:V) {
    if(R[s,i] != -1) a = c(a, i)
  }
  return(a)
}
```

```
while (r <= rounds) {
  s = sample(S, 1)
  while (TRUE) {
    action_space = get_actions(s)
    action <- sample(action_space, 1)
    s_next <- action
    actions_next = get_actions(s_next)
    qs = c()
    for (i in actions_next) qs = c(Q[s_next,i])
    Q[s,action] <- R[s,action] + alpha * max(qs)
    s = s_next
    if (s == END) break
  }
  r <- r+1
}
Q
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
##  [1,]  0.00 33.75  0.00 33.75  0.00  0.00  0.00  0.00   0.0
```

```
##  [2,] 20.25  0.00 56.25  0.00 56.25  0.00  0.00  0.00   0.0
##  [3,]  0.00 33.75  0.00  0.00  0.00 93.75  0.00  0.00   0.0
##  [4,] 20.25  0.00  0.00  0.00 56.25  0.00 56.25  0.00   0.0
##  [5,]  0.00 33.75  0.00 33.75  0.00 93.75  0.00 93.75   0.0
##  [6,]  0.00  0.00 56.25  0.00 56.25  0.00  0.00  0.00 156.2
##  [7,]  0.00  0.00  0.00 33.75  0.00  0.00  0.00 93.75   0.0
##  [8,]  0.00  0.00  0.00  0.00 56.25  0.00 56.25  0.00 156.2
##  [9,]  0.00  0.00  0.00  0.00  0.00 93.75  0.00 93.75   0.0
```

```
path = c()
state = 1
Q[Q == 0] <- 1000
while (length(path) < V)
{
  pre_state = state
  path = c(path, state)
  state = match((min(Q[state,])), Q[state,])
  Q[pre_state, ] = 1000
  Q[, pre_state] = 1000
}
path
```

```
## [1] 1 2 3 6 5 4 7 8 9
```