NOC AI SMPS 2021

(Version 0.1)

We will use the following version of Beam Search in assignments and in the final exam. It exhibits assignment-friendly termination property that avoids infinite loops.

```
BEAM-SEARCH(S, w)
 1  OPEN ← S : []
 2  N ← S
 3  do  bestEver ← N
 4      if GOALTEST(OPEN) = TRUE
 5          return goal from OPEN
 6      neighbours ← MOVEGEN(OPEN)
 7      OPEN ← take w (sort_h neighbours)
 8      N ← head OPEN      ▷ best in new layer
 9  while h(N) is better than h(bestEver)
10  return bestEver
```

Often, to find an upper bound U, we use the following version of Beam Search algorithm that expands every layer. If the goal node is not reachable, for some state spaces, it may loop forever. So use with caution.

```
BEAM-SEARCH(S, w)
1  OPEN ← S : []
2  while GOALTEST(OPEN) = FALSE
3      neighbours ← MOVEGEN(OPEN)
4      OPEN ← take w (sort_h neighbours)
5  return goal from OPEN
```

Usage of **take** function.

$[o, u, t] = \textbf{take}\ 3\ [o, u, t, r, u, n]$

$[a, t] = \textbf{take}\ 3\ [a, t]$

$[a] = \textbf{take}\ 3\ [a]$

$[] = \textbf{take}\ 3\ []$