Problem 0



|  |  |
| --- | --- |
| States | Single word unigrams; a single letter in the input query |
| Actions | Insert vowel, insert space |
| Cost | Fb(action) |
| Start State | “-BEGIN-“ keyword so we can start before index 0 |
| End State | Last letter in query |

H(s) is consistent because the cost for adding a vowel or space always <= adding a whole new word with a space.

1. BFS is a special case of UCS, which is a special case of A\*.
2. Writing out potential values of n:

n = 0: length of query is 0 -> states\_expanded = 0

n = 5: length of query is 5 -> start, \_, \_, \_, end -> states\_expanded = 5

…

Therefore upper-bound = O(n).

1. The two answers are not similar, as mine was not exponential. This could be because mine did not consider different options for adding vowels or spaces. I only considered either we add a vowel or space, or we don’t. But I did not consider if we could have different options for adding vowels and spaces (e.g. adding multiple vowels).

Problem 1

1. Vminmax(s, d) = {

Gamevalue(s); if isend(s)

Eval(s); if d = 0

Maxa in A(s) Vminmax(next(s, a), d); if player(s) == max

Mina in A(s) Vminmax(next(s, a), d-1); if player(s) == min

}

Problem 3

1. Vminmax(s, d) = {

Gamevalue(s); if isend(s)

Eval(s); if d = 0

Maxa in A(s) Vminmax(next(s, a), d); if player(s) == max

∑a in A(s) πopp(s,a) Vminmax(next(s, a), d-1); if player(s) == min

}