CS221 - Homework 3
(Reconstruct)

Name: Vinod Kumar Senthil Kumar
SUNET ID: vinodkum
Contributors: Xun Wang, Joe Fan

1a) Let the input be

"is land travel"

Let the costs ~~function~~ be,

$c(\text{-begin-}, \text{is}) = 1$

$c(\text{-begin-}, \text{is land}) = 2$

$c(\text{is}, \text{land}) = 2$

$c(\text{is land}, \text{travel}) = 1$

$c(\text{land}, \text{travel}) = 1$, otherwise cost $= \infty$

By greedy approach, the output would be

"is land travel"

Total cost $= 1 + 2 + 1 = 4$

But using DP or UCS or A^* would give the output,

"island travel"

Total cost $= 2 + 1 = 3$

thus proving that the greedy approach is sub-optimal.

2a) Let the input be

['wht', 'brd']

Let the possible words be

possibleFills('wht') = ['wheat', 'white']

possibleFills('brd') = ['bird']

Let the bigram costs be

c(-begin-, wheat) = 1

c(-begin-, white) = 2

c(wheat, bird) = 3

c(white, bird) = 1, otherwise cost = ∞

By greedy approach, the output would be

"wheat bird"

Total cost = 1 + 3 = 4

But using DP or UCS or A* would give us

"white bird"

Total cost = 2 + 1 = 3

thus proving that the greedy approach is sub-optimal.

3a) The search problem is formulized as follows:

i) State :

It is a tuple of index of the character in the input to process next & the previous word.

(index of character to process next, last word)

ii) Action :-

It denotes a segmented, vowel ~~filled~~ ~~word~~ inserted word possible from the current state. It denotes our choice of a word from the ~~possi~~ list of possible words given by possibleFills()

iii) Cost :

It is the bigram cost of our choice of segmented, vowel-inserted word (i.e. action) preceeded by the previous word from the state.

b (our choice from possibleFills(), previous word from state)

iv) Initial state:

The initial state would be

~~0, words~~

(0, -begin-)

v) goal test:

We have reached goal if there are no more characters to process. In other words, if the last processed character's index from the state equals the index of the last character in the input, we have reached goal.

state[0] == len(input query)

3c) I define $u(w)$ as the minimum of bigram costs of the word 'w' preceeded by all the words in the corpus.

$$u(w) = \min_{w' \in \text{corpus}} b(w', w)$$

Let the heuristic $h_u(s)$ be defined as the minimum unigram cost of all possible actions from state 's'. An action from state 's' denotes a segmented, vowel inserted word possible from the state 's'.

$$h_u(s) = \min_{a \in \text{actions from state 's'}} u(a)$$

Intuitively, $h_u(s)$ gives the lowest cost action possible from the state 's', which in turn strives to reach the goal with minimal cost.

To prove that the heuristic is consistent,

i) we need to prove that

$$cost'(s,a) = cost(s,a) + h_u(succ(s,a)) - h_u(s) \geq 0$$

As ~~the~~ $h_u(s)$ is a unigram cost, which in turn is a bigram cost which is non-negative, $h_u(s)$ is non-negative ~~&~~ (i.e $h_u(succ(s,a)) \geq 0$)
Also as $h_u(s)$ gives the minimum unigram cost, which in turn is the minimum bigram cost, ~~$h_u(succ(s,a))$~~ $h_u(s)$ will be the lower bound of $cost(s,a)$

In other words, $cost(s,a) \geq h_u(s)$

$$cost(s,a) - h_u(s) \geq 0$$

Thus,

$$cost'(s,a) = \left(cost(s,a) - h_u(s)\right) + h(succ(s,a))$$

$$= \text{non-negative} + \text{non-negative}$$

$$= \text{non-negative}$$

$cost'(s,a) \geq 0$ (proved)

ii) Heuristic of goal state must be zero.

At the goal state, there won't be any possible segmented, ~~vowel~~ vowel-filled word (i.e action). Thus the cost will be zero.

$$h_u(\text{goal state}) = \min_{\substack{a \in \text{action} \\ \text{at goal} \\ \text{state}}} u(a)$$

$$= \min_{a \in \{\}} u(a)$$

$$= 0$$

As both conditions are satisfied, the heuristic $h_u(s)$ is consistent.