```
function [category_index predicted_vector final_delta min_difference
predicted_vector_array] = predict_best_fit_tree_N(anchor_tree, delta_tree,
new_data_item, predict_toggle, search_toggle, missing_data_vector, N)

%search_toggle must be -1,0,1,2;
%this determines whether the algorithm terminates upon the first match, best match
at first depth, or best match in the entire tree (-1,0,1), respectively.
%Search toggle of 2 switches off the possibility of a fail.
%The missing_data_vector contains the indeces of the missing data items, and is
ignored unless predict_toggle = 1.
%predicted_vector will be the best match that satisfies the search_toggle criteria.
%The predicted_vector_array will contain all matches generated during the search.

[width depth] = size(anchor_tree);
new_data_vector = new_data_item{1};

predicted_vector_array = {};
index = 1; %this is the index for the predicted_vector_array

%this is to ensure we ignore any labels in the data
new_data_vector = new_data_vector(1:N);

%We initialize the return values to indicate a failure to categorize
min_difference = Inf;
final_delta = Inf;

i = 0;
break_loop_depth = 0;
break_loop_width = 0;
match = 0;

while(i < depth && break_loop_depth == 0)

  j = 0;

  while(j < width && break_loop_width == 0)

    anchor_vector = anchor_tree{width - j, depth - i};
    delta = delta_tree{width - j, depth - i};

    if(isempty(anchor_vector) == 0)

      if(predict_toggle == 1)

        new_data_vector(missing_data_vector(:)) =
anchor_vector(missing_data_vector(:));

      endif
```

```
      temp_anchor = anchor_vector(1:N);

      difference = norm(new_data_vector - temp_anchor);

      %Either we have a match, or we've reached the top of the tree, or we've
disabled fails
      if(difference <= delta || anchor_vector(3) == Inf || search_toggle == 2)

        %if true, then we have a match
        if(difference < min_difference)

          match = 1;
          predicted_vector = new_data_vector;
          category_index = [(width - j) (depth - i)];
          final_delta = delta;
          min_difference = difference;

          %if true, then we terminate upon first match
          if(search_toggle == -1)

            break_loop_depth = 1;
            break_loop_width = 1;

          %if true, then we terminate at the end of the current depth
          elseif(search_toggle == 0)

           break_loop_depth = 1;

          %otherwise, search_toggle is 1 or 2, so we search the entire tree

          endif

        %if true, then we've reached the top of the tree with no match
        elseif(anchor_vector(3) == Inf && match == 0)

          predicted_vector = anchor_vector;
          category_index = [(width - j) (depth - i)];

        endif

        %In all cases except where we've reached the top of the tree, we have a
possible match
        if(anchor_vector(3) != Inf)

          predicted_vector_array{index} = new_data_vector;
          index = index + 1;
```

```
        endif
      endif
    endif
    j = j + 1;
  endwhile
  i = i + 1;
endwhile
endfunction
```