

CMSC425 Spring 2019

Homework 3: Shapes, animation and intelligence

Assigned Tuesday, April 24th

Due by midnight on Sunday, May 5th

Submit PDF on Elms (no paper required)

Part a. Warm up problems

Again, these are intended as straightforward use of basic formulas to review and debug your understandings of the concepts.

1. The Unity mechanim animation system has a feature that can set the joint angles of a humanoid model to cause it to turn its head to face a particular direction. (Really!) This would best be described is an example which of the following animation techniques (select one):

- (i) Keyframe animation
- (ii) Motion capture
- (iii) Inverse kinematics

2. Our proposed algorithm for triangulating the walkable region in the construction of navigation meshes repeatedly cut off the ear such that the cutting edge has minimum length. What is an ear of a simple polygon? What is the reason for favoring short cutting edges?

3. In our description of the four elements of the boid model for flocking behavior (separation, alignment, avoidance, and cohesion), what is the purpose of alignment, and how might it be implemented?

4. Behavior trees have two types of task nodes, sequences and selectors. In a sequence node, its children are evaluated from left to right, and each returns either success or failure. Under what circumstances does the sequence node itself return success? Under what circumstances does a selector node return success.

Part b. Applications

1. **Winged edge meshes.** Present a procedure (in pseudocode) that, given a half edge e of the DCEL, returns a list L consisting of the vertices that are adjacent to either of e 's endpoints. The vertices should be listed in counterclockwise order about e . The list can start with any vertex, and duplicates are allowed.

For example, given the example shown in Fig. 1, the list $L = \langle v_0, v_1, \dots, v_6 \rangle$ would be one valid result (as would any cyclical shift of this sequence). Your procedure should run in time proportional to the length of the output. (Hint: The answer is simpler if you choose the starting point carefully.)

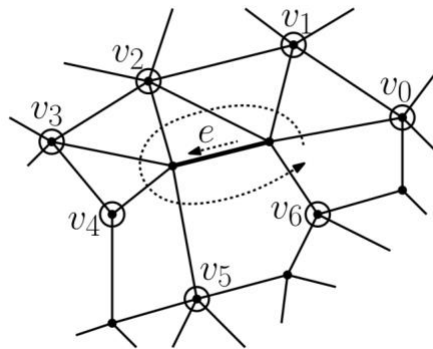


Figure 1

2. **A* search.** In this problem, we will consider A* search under both admissible and inadmissible heuristics.

Consider the graph shown in Figure 2. For each node u , define $\text{dist}(u, t)$ to be the *straight-line distance* from u to t . For example, $\text{dist}(s, t) = 8$ and $\text{dist}(c, t) = 2$.

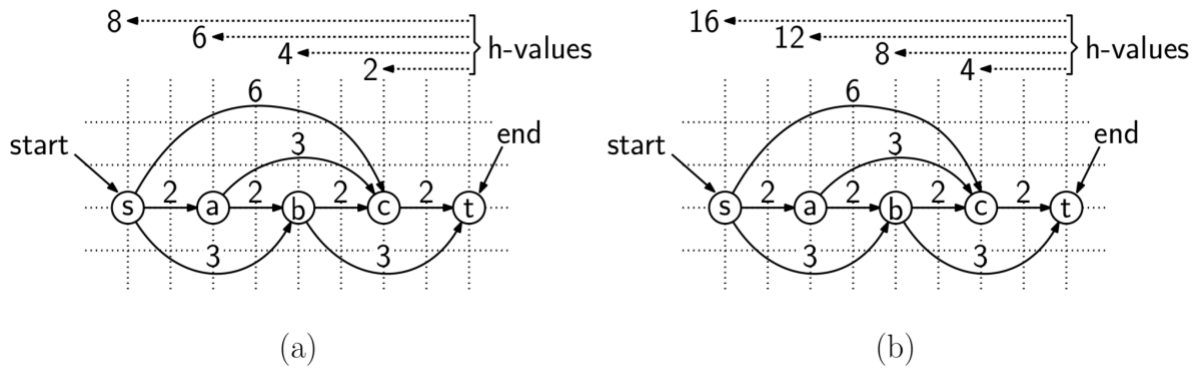


Figure 2.

ERRATA: the distance function $\text{dist}(u, t)$ is not admissible in example (a). Why?

If you have done the problem with the original data, submit it as given.

If you have not, for (a) use the new heuristic $h'(u) = \text{dist}(u, t)/2$. Eg, halve the dist function.

(a) Suppose that we take the admissible heuristic $h(u) = \text{dist}(u, t)$ (see Fig. 3(a)). (For example, $h(s) = 8$, $h(c) = 2$ and $h(t) = 0$.) Trace the execution of the A* algorithm on this graph using s as the start and t as the destination. In particular:

- list the nodes as they are processed, indicating the values of $d[u] + h(u)$
- whenever a node is processed, indicate how the d -values of its neighbors are updated
At the end (when t is processed), show the final d -values are for all the nodes.

(b) Suppose that we take the inadmissible heuristic $h(u) = 2 \cdot \text{dist}(u, t)$ (see Fig. 3(b)). (For example, $h(s) = 16$, $h(c) = 4$ and $h(t) = 0$.) Repeat (a) but using this different value of h .

(c) Did the algorithm produce the correct answer in part (b)? Explain briefly.