The $A^*$ Algorithm

We walk through an example computation of the $A^*$ algorithm for solving the single pair minpath problem on a weighted directed graph. The pair is $(S, T)$. Arc weights are shown as black numerals, we write $w(x, y)$ for the weight of the arc from $x$ to $y$.

![Graph Diagram](image)

The heuristic $h(x)$ for each vertex $x$ is indicated by a red numeral.

![Heuristic Graph Diagram](image)

Just as for Dijkstra’s algorithm, we maintain three sets of vertices: fully processed, indicated by a blue background, partially processed, indicated by a green background, and unprocessed, indicated by no background. The partially processed vertices are held in an updatable minqueue.

For each fully or partially processed vertex $x$, we let $f(x)$ be the length of the shortest path so far found, indicated by a blue numeral.

We let $g(x) = f(x) + g(x)$, indicated by a green numeral.

Initially, there are no fully processed vertices, and only the source vertex $S$ is partially processed.
At each step, if $g(x)$ is the minimum value over all partially processed vertices, $x$ becomes fully processed, and all its unprocessed out-neighbors become partially processed. During this step, $S$ becomes fully processed, and $A$ and $H$ become partially processed.

At this step, $A$ becomes fully processed, while $B$, $D$, and $E$ become partially processed. Backpointers are indicated as red arrows.

$E$ becomes fully processed, while $F$ becomes partially processed.
B becomes fully processed, while C becomes partially processed.

C becomes fully processed. D acquires a new, smaller value of $f$, and its backpointer changes to C.

D and G become fully processed, while T becomes partially processed.
It seems unnecessary, but the algorithm only stops when $T$ becomes fully processed. Although not in this example, it is possible that $T$ would acquire a new backpointer after being partially processed for the first time.

Errors fixed. If you detect another error, please send me email as soon as possible.