Human Cannonball Run

There are lots of human cannons available. You run at 5m/s and all cannons launch you 50m in 2s. Given your location, a destination, and the positions of cannons, can you calculate the minimal time to travel?

Possible Paths

Starting point -> destination

Possible Paths

Starting point -> destination
Starting point -> cannon -> destination

Possible Paths

Starting point -> destination

Starting point -> cannon -> destination

Starting point -> cannon 1
-> cannon 2 -> ... -> cannon N
-> destination

Timing

Starting point -> cannon or destination

- time = distance / 5

Timing

Starting point -> cannon or destination

    time = distance / 5

Cannon -> cannon or destination

    time = 2 + |distance - 50| / 5

Timing

Starting point -> cannon or destination

  time = distance / 5

Cannon -> cannon or destination

  time = 2 + |distance - 50| / 5

(You will never run to a cannon but not use it. Why?)

Problem Modeling

Cannon = vertex

Starting point, destination = vertex

Run / launch = edge

Time = edge length

Solution

"__ ________!"

Which solution to use mainly depends on the maximum number of cannons.

Solution

For a small number of cannons:

  Floyd -> $O(n^3)$

Larger:

  Dijkstra -> $O(n^2)$

  (Possible optimization: A*)

Dijkstra's Algorithm

- Setup a queue Q
- Setup a minimum distance table D
- Push the starting point into Q
- Perform the iteration on Q until the destination is popped

Dijkstra Iteration

- Pop the vertex V with smallest D_V from Q
- For all vertices U that an edge $E_{V \rightarrow U}$ exists
    - $D_U = \min(D_U, D_V + |E_{V \rightarrow U}|)$
    - If D_U changed, insert/move U -> Q

By the way...

Dijkstra Iteration

- Pop the vertex V with smallest D_V from Q
- For all vertices U that an edge $E_{V \to U}$ exists
    - $D_U = \min(D_U, D_V + |E_{V \to U}|)$
    - If D_U changed, insert/move U -> Q

??? Iteration

- Pop the vertex V with smallest D_V from Q
- For all vertices U that an edge $E_{V \to U}$ exists
    - $D_U = \min(D_U, ???)$
    - If D_U changed, insert/move U -> Q

Prim Iteration

- Pop the vertex V with smallest D_V from Q
- For all vertices U that an edge $E_{V \to U}$ exists
    - $D_U = \min(D_U, |E_{V \to U}|)$
    - If D_U changed, insert/move U -> Q

A* Iteration

- Pop the vertex V with smallest D_V from Q
- For all vertices U that an edge $E_{V \rightarrow U}$ exists
    - $D_U = \min(D_U, D_V + H_U + |E_{V \rightarrow U}|)$
    - If D_U changed, insert/move U -> Q