# Assignment 5 : CSL 201
# Topic : Robot Motion Planning

In this assignment, we will implement Stentz's algorithm for path finding by a robot in unknown terrain. We will assume that the robot's environment is a *weighted, undirected* graph $G(V, E)$. The robot is initially located at a vertex $s \in V$ and its goal is to reach a vertex $t \in V$ in the graph.

If the terrain graph was known completely by the robot, it can reach the goal by computing the shortest path from $s$ using Dijkstra's algorithm. However, the catch is that some subset $B \subset V$ of vertices in the terrain graph are *blocked*. In other words, there are obstacles at some vertices which make those vertices impassable. Thus, a robot can follow a path to the goal which goes only through unblocked vertices.

To infer the presence of unknown obstacles, the robot is equipped with what are called *tactile sensors*. Tactile sensors are the weakest form of sensors - a robot with tactile sensors can detect an obstacle only by "touching" it.

To be more precise, suppose $(u, v)$ is an edge of the graph, the robot is currently at vertex $u$, and vertex $v$ is blocked. Further, assume that the robot is unaware of an obstacle at vertex $v$. Then, the robot starts moving along the edge from $u$ to $v$, till it reaches $v$, at which point its tactile sensors sense an obstacle, and it becomes aware of an obstacle at $v$.

We assume that the robot is always aware of its location within the graph $G(V, E)$. This can be ensured by equipping it with two devices : (i) an odometer, and (ii) a compass. An odometer measures the distance traveled by the robot, and a compass measures the direction of travel. By combining the readings of both instruments, the robot can exactly trace out the path it is following, as well as its current location in the graph.

The robot can solve this path planning problem using Stentz's algorithm, which uses Dijkstra's algorithm as a subroutine. Suppose the terrain is $G(V, E)$, the set of blocked vertices is $B$, the start vertex is $s$ and the goal vertex is $t$.

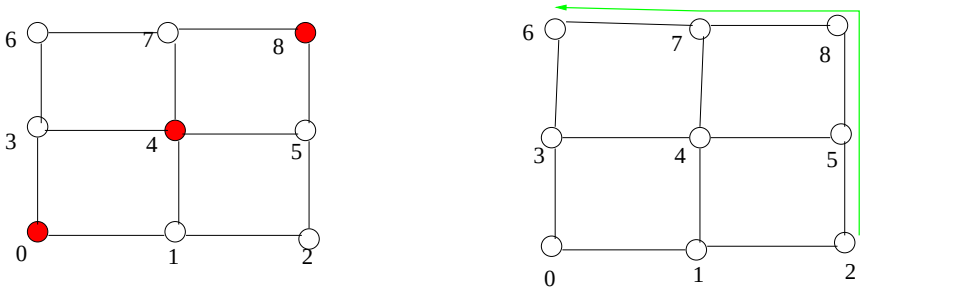
The robot maintains two parameters : (i) its current location $v$, and (ii) the set of blocked vertices $S$ discovered till now. Initially, the robot is aware of no blocked vertices, and hence $S = \phi$. Stentz's algorithm is the following:

1. Compute a shortest path $P$ from the current vertex $v$ to the goal $g$ in graph $G(V, E)$ with all vertices in $S$ blocked.
2. If there is no such path, output "No path to goal exists" and stop.
3. Otherwise, start following path $P$ to the goal till you either (i) reach the goal, or (ii) find a new blocked vertex $b \in P$ on the path.
4. In case (i), output "Reached goal" and stop. Otherwise, add vertex $b$ to the set $S$ of vertices discovered to be blocked so far, set the current vertex $v$ to be the vertex preceeding $b$ on path $P$, and go to step 1.

The following figures show 4 iterations of the above algorithm on a grid graph with 9 vertices labeled from 0 to 8. The robot is initially at vertex 2 and

the goal is at vertex 6. All edges have weight 1. Three vertices $0, 4$, and 8 are blocked, but the robot is unaware of any blocked vertices. Thus, the initial map of the robot assumes that all locations are traversable.

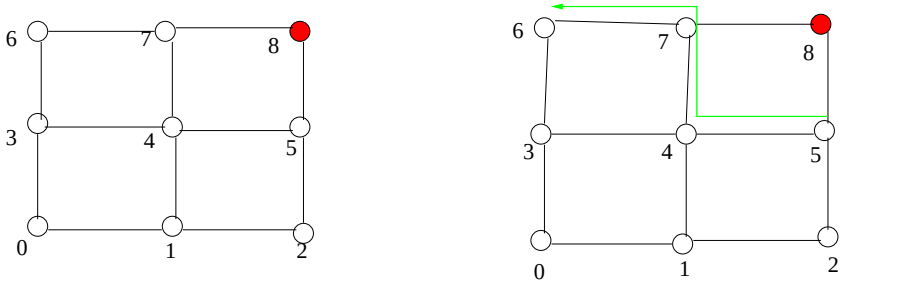
In the first step, the robot sees its initial map, and finds a path from 2 to 6 of cost 4. The robot starts moving on this path, but detects a blocked vertex at 8. The robot next updates its map by making 8 blocked, and recomputes a shortest path from its current location 5 to the goal. This time, it comes up again with a path (5 4 7 6) of length 4. It starts following this path, but finds a blocked vertex at 4. It updates its map which now has two blocked vertices 4 and 8, and recomputes a path (5 2 1 0 3 6) of length 5 from its current location to the goal. On following this path, the robot hits a blocked vertex at 0, and updates its map accordingly. Next, it tries to find a shortest path from its current location 1 to the goal, but no such path exists. Hence, it stops at vertex 1 and outputs that "No path to goal exists".




Actual Terrain

Robot's Initial Map with a shortest path from 2 to 6

Figure 1: Robot path planning

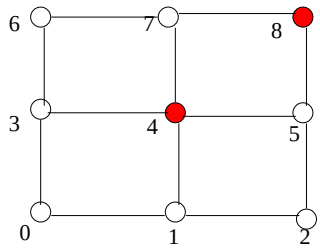



Robot's Map after finding a blocked vertex at 8

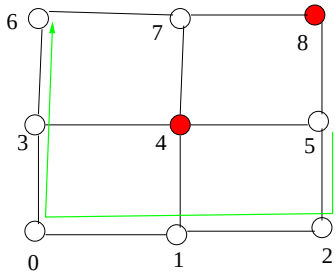
A new shortest path from current location 5 to goal

Figure 2: Robot path planning

In this assignment, you will be a given as input an undirected, weighted graph $G(V, E)$, a start vertex $s$, a goal vertex $t$, and the set $B \subset V$ of blocked

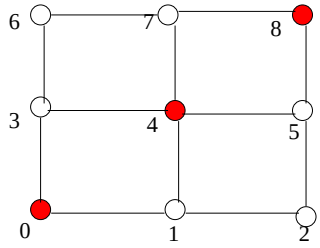



Robot's Map after finding a blocked vertex at 4

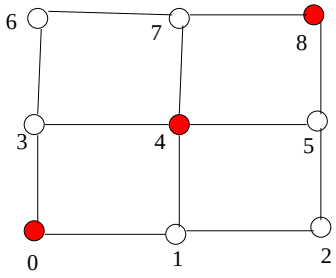



A new shortest path from current location 5 to goal

Figure 3: Robot path planning




Robot's Map after finding a blocked vertex at 0




No shortest path from current location 1 to the goal

Figure 4: Robot path planning

vertices. You have to output the behavior of Stentz's algorithm assuming that the robot is initially unaware of any blocked vertices. The input and output format is described in the following sections.

# 1 Input Format

The first line of input contains a single number $n$, which is the total number of vertices in graph $G$. The vertices of $G$ are assumed to be labeled from 0 to $n-1$. The second line contains two numbers $s$ and $t$, which correspond to the start and goal vertices respectively.

After this, the adjacency list representation of graph $G$ is given on the next $n$ lines, with the $i$th line giving the adjacency list for vertex $i-1$. Note that since $G$ is weighted, the adjacency list for each vertex stores weight of the edge in addition to the number of the adjacent vertex.

If vertex $i-1$ has a total of $k$ edges to vertices $j_1, j_2, \cdots, j_k$ with weights $w_1, w_2, \cdots, w_k$ respectively, line $i$ will contain the number $k$ of edges, followed by $k$ pairs of integers, each representing the next vertex and weight of an edge incident on vertex $i$. In this case, line $i$ will be:

$$k\ j_1\ w_1\ j_2\ w_2 \cdots\ j_k\ w_k$$

After the $n$ lines giving the adjacency lists of vertices $0, 1, \cdots, n-1$, there is a single integer $b$ on a line by itself, which is equal to the number of blocked vertices in $G$. The next line containes $b$ numbers, which are the indices of the blocked vertices in any order.

For example, the input for path finding problem on the 9-node grid graph described above will be:

```
9 // total number of vertices in the graph
2 6 // the start and goal vertices
2 3 1 1 1
3 4 1 2 1 0 1
2 5 1 1 1
3 6 1 0 1 4 1
4 7 1 1 1 5 1 3 1
3 8 1 2 1 4 1
2 3 1 7 1
3 4 1 8 1 6 1
2 5 1 7 1
3 // number of blocked vertices
0 4 8 // the set of blocked vertices
```

# 2 Output Format

The output of your program will succintly describe the behaviour of Stentz's algorithm. Every time the algorithm finds a new path $P = (v, w_1, w_2, \cdots, w_k, t)$ of length $l$ from the current vertex $v$ to goal $t$, the algorithm will print:

$$Found\ new\ path\ from\ v\ of\ length\ l:$$

, followed by the sequence of vertices from the current vertex to the goal in path $P$:

$$v\ w_1\ w_2 \cdots\ w_k\ t$$

If the robot finds a new blocked vertex $b$, the algorithm will print a single line:

$$Found\ blocked\ vertex\ at\ b$$

If the goal is not reachable, the algorithm will print a single line:

$$No\ path\ to\ goal\ exists$$

, and stop. If the goal is reached by the robot, the algorithm will print a single line:

*Goal reached*

, and stop.

For example, here is sample output for the 9-node grid graph described above:

```
Found new path from 2 of length 4 :
2 5 8 7 6
Found blocked vertex at 8
Found new path from 5 of length 3 :
5 4 7 6
Found blocked vertex at 4
Found new path from 5 of length 5 :
5 2 1 0 3 6
Found blocked vertex at 0
No path to goal exists
```

Note that your outputs may differ from the standard output as there may be several shortest paths possible from the current vertex to the goal vertex. This is fine, as long as your output exactly follows the above description. The correctness of your output will be checked by a program.

The file "test_cases" describing the test cases for the assignment along with 7 input files $in0, in1, \cdots, in6$ is given in the tar archive "input-assgn5.tar.gz".

# 3 Suggestions

1. The simplest algorithm is the following: everytime a new blocked vertex is discovered, remove it from the vertex set of graph $G$, and run Dijkstra's algorithm from the current vertex $v$ to find the next shortest path to the goal.

2. However, the above algorithm is expensive, and this is not how Stentz's algorithm is implemented. There are two tricks which make it run faster:

   (a) Rather than computing the shortest path from the current vertex to goal, always compute the shortest path "backwards" with the goal as the source vertex.

   (b) This has the added advantage that when a vertex $b$ is found to be blocked, only the distances of vertices in $b$'s subtree in the shortest path tree already computed can change. The remaining vertices remain unaffected.

   These improvements to the basic algorithm will be discussed in lab timings.