Lección 5: Planeación de Movimientos

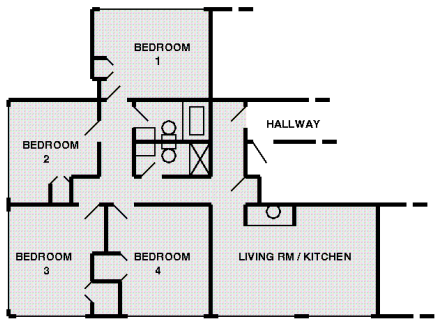
Jesús Savage
Facultad de Ingeniería, UNAM
Trabajo realizado con el apoyo del Programa
UNAM-DGAPA-PAPIME PE100821

19 de junio de 2023

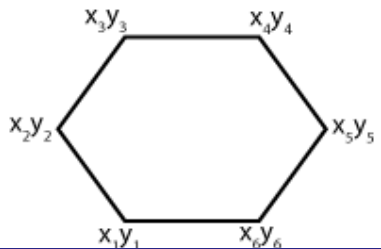
Índice

Introducción

En los modelos tradicionales de los robots móviles se tiene una representación del medio ambiente:



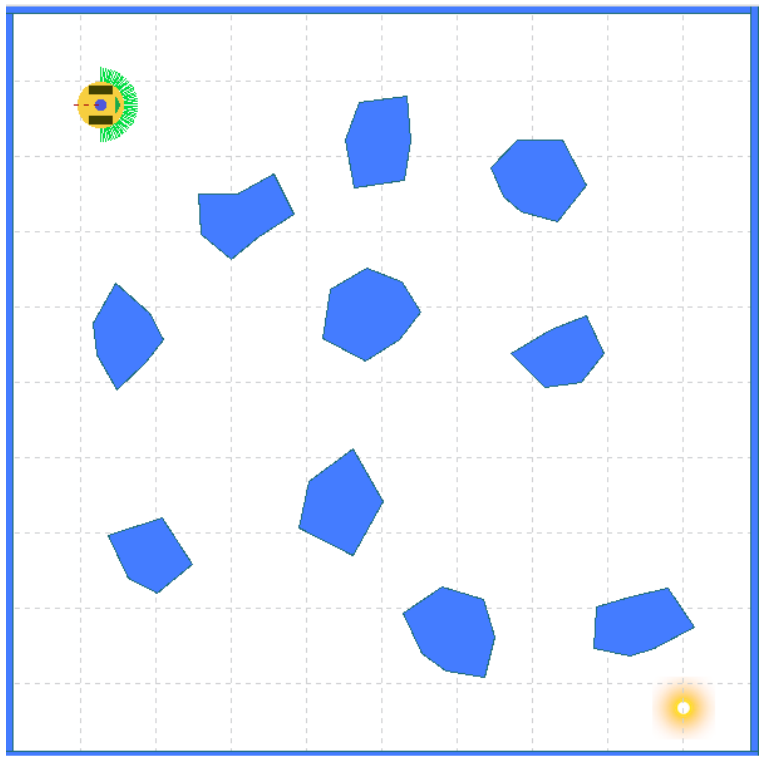
Con una representación simbólica de los objetos en cada cuarto, éstos se representan por medio de poligonos en donde se tienen su vertices Xi, Yi, ordenados en el sentido al de las manecillas del reloj. Estos poligonos separan el espacio ocupado y el espacio libre en donde puede navegar el robot.



Planeación de Movimientos

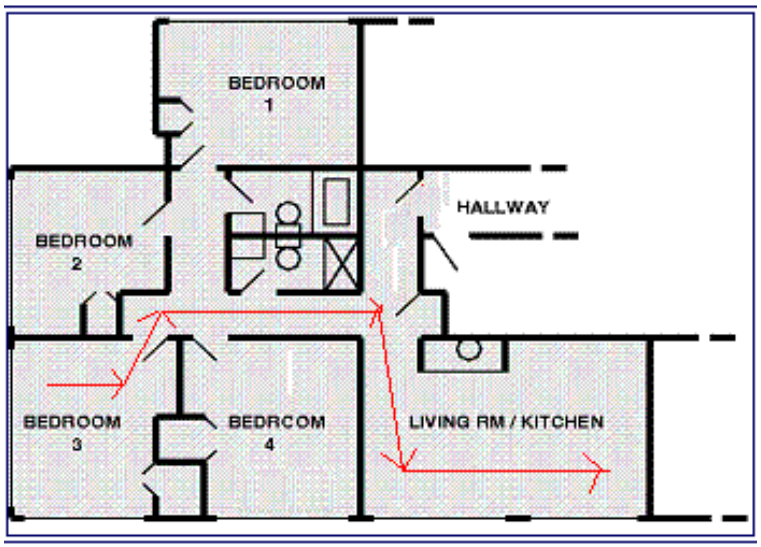
Objetivos:

- Encontrar rutas posibles de moviento del robot de un punto origen a un punto destino.
- Escoger la mejor de estas rutas con base a un criterio de optimización.



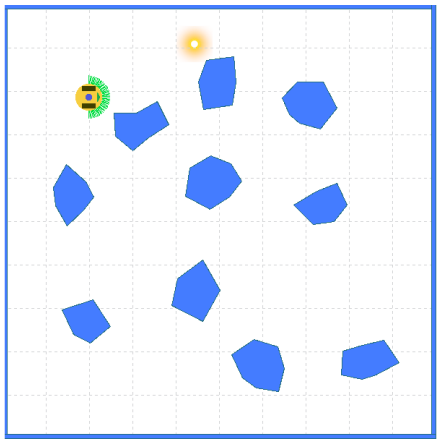
Introducción Planeación de Movimientos

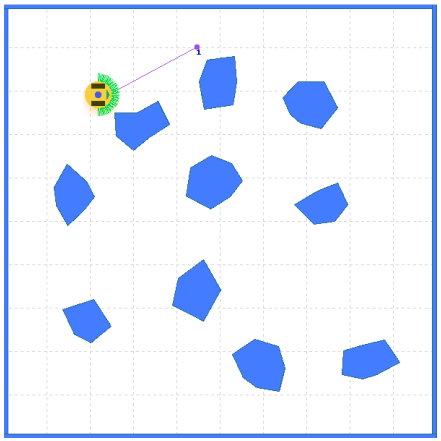
Si el punto inicial y el objetivo están en regiones diferentes, entonces se tiene que encontrar primero una ruta global entre ellos y después se encuentra para cada región una ruta local óptima.



Introducción Planeación de Movimientos

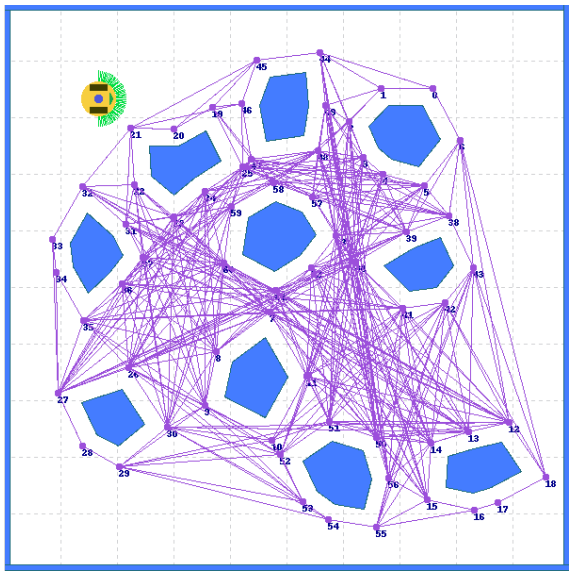
Partiendo del cuarto en donde se encuentra el robot se traza una línea recta al punto destino de salida de ese cuarto y sino se encuentra otro objeto que esté estorbando entonces hay un camino del origen al destino.





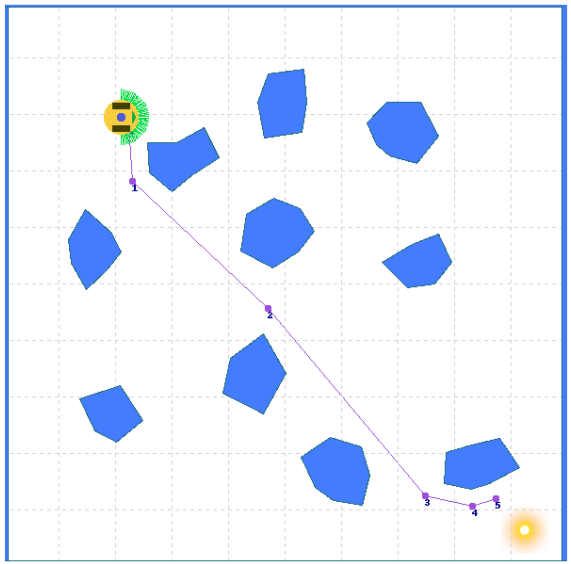
Introducción Planeación de Movimientos

Si no hay una ruta directa entonces se buscan rutas opcionales (indirectas), mediante trayectorias que se encuentran usando como puntos intermedios las esquinas de los polígonos extendidos que obstruyen el paso.



Introducción Planeación de Movimientos

Al final de este proceso se contará con un conjunto de rutas opcionales de las cuales se escogerá la mejor de acuerdo a ciertos criterios.



Búsqueda de la mejor ruta

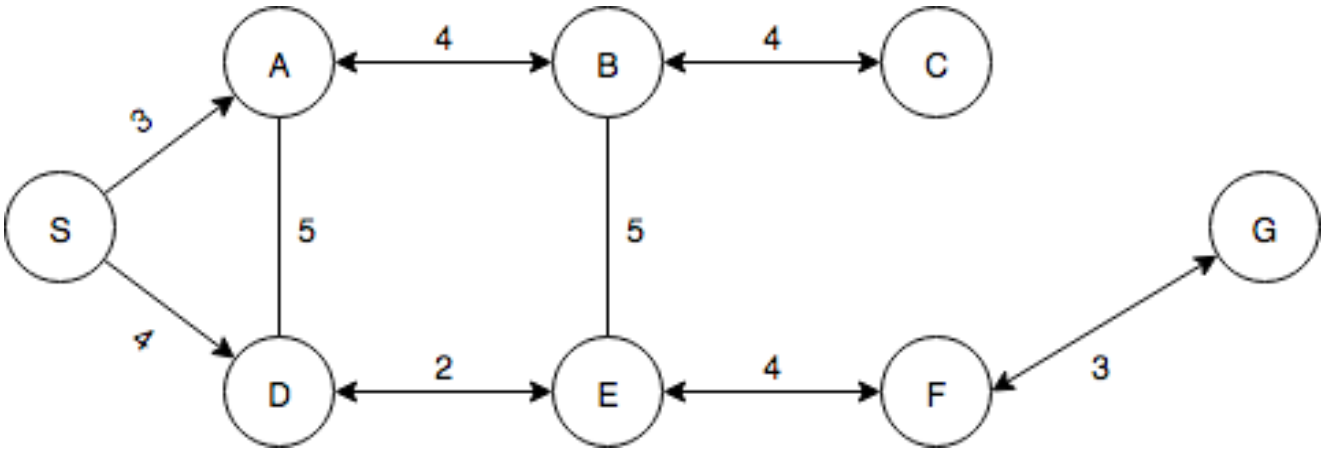
El problema de la búsqueda es, dado:

1. Un punto inicial (o nodo)
2. Un punto objetivo (o nodo)
3. Un mapa de nodos y conexiones

Objetivo:

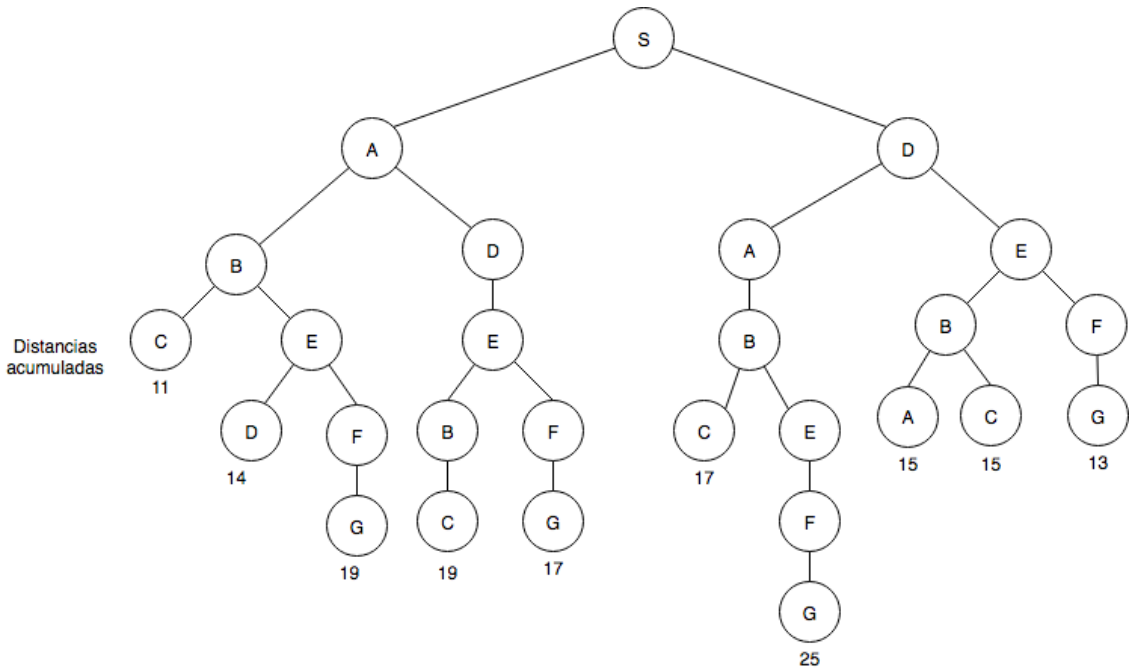
1. Encuentra una ruta o la mejor ruta que encuentre el punto objetivo.
2. Recorra la ruta

Una colección de nodos y conexiones entre ellos es llamado una Red.

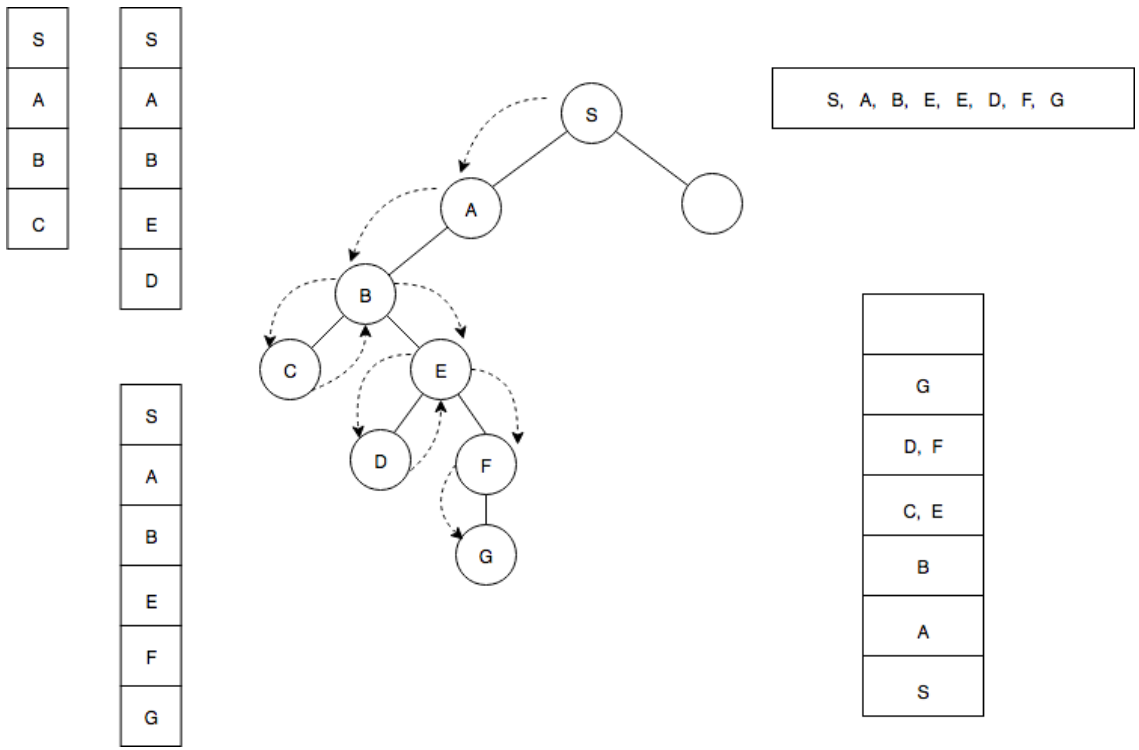


Búsqueda de la mejor ruta

Una ruta es encontrada empezando desde el nodo del principio S y terminando en el nodo objetivo G. Procedimientos de búsqueda exploran redes como estas para encontrar caminos que unan el origen con el destino. Una colección de nodos y ramas es llamado un árbol. A continuación se observa un árbol formado por la red anterior.

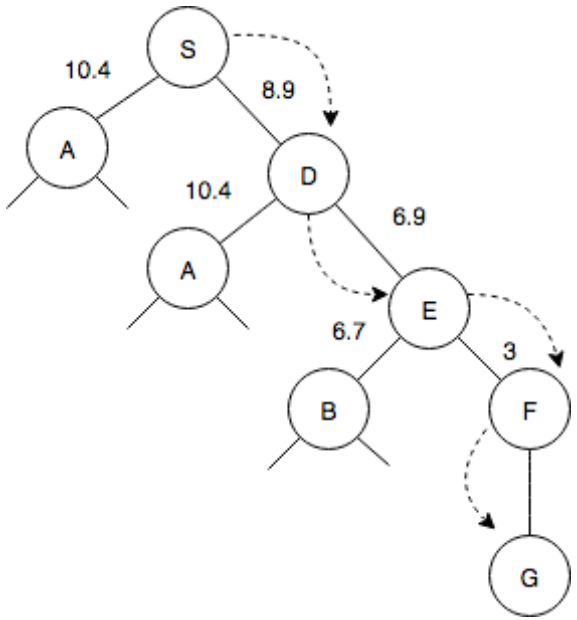


Búsqueda Primera en Profundidad (Depth First Search)

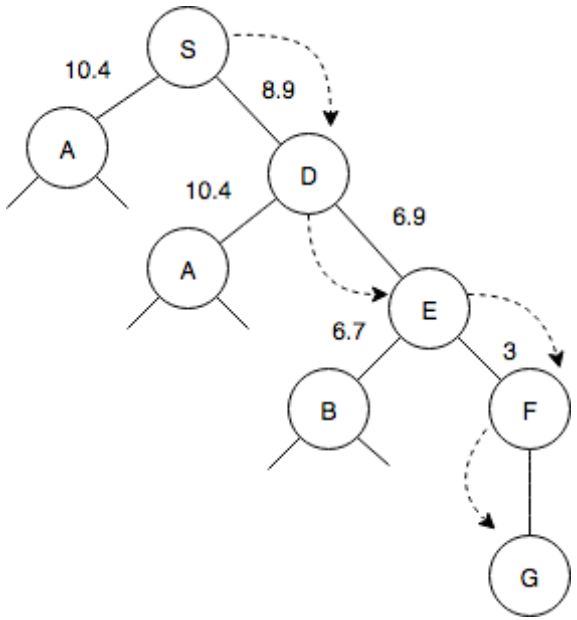


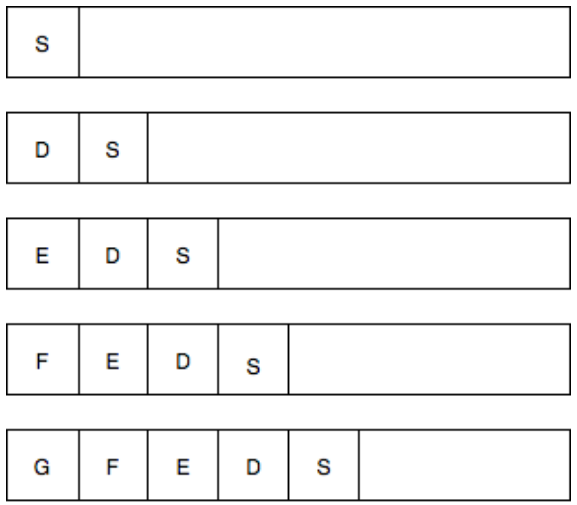
Método Subiendo el Monte (Hill Climbing)

Es igual a búsqueda primera en profundidad pero se usa una función heurística la cual indica cual de los nodos seguir. La función heurística para este problema puede ser la distancia euclidiana de la posición del nodo en el medio ambiente y la posición del destino.

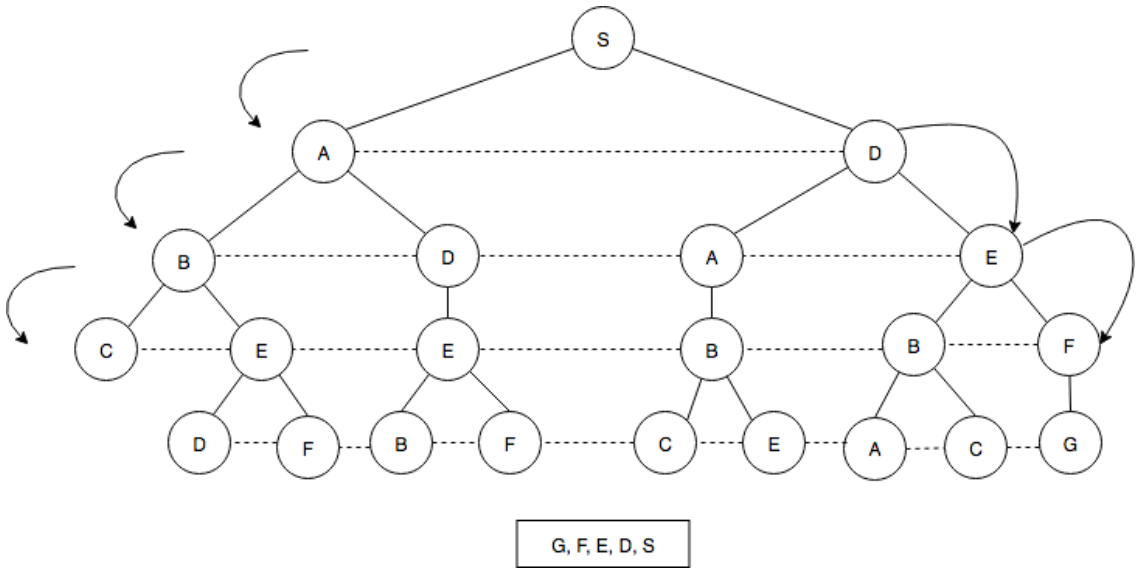


Método Subiendo el Monte (Hill Climbing)

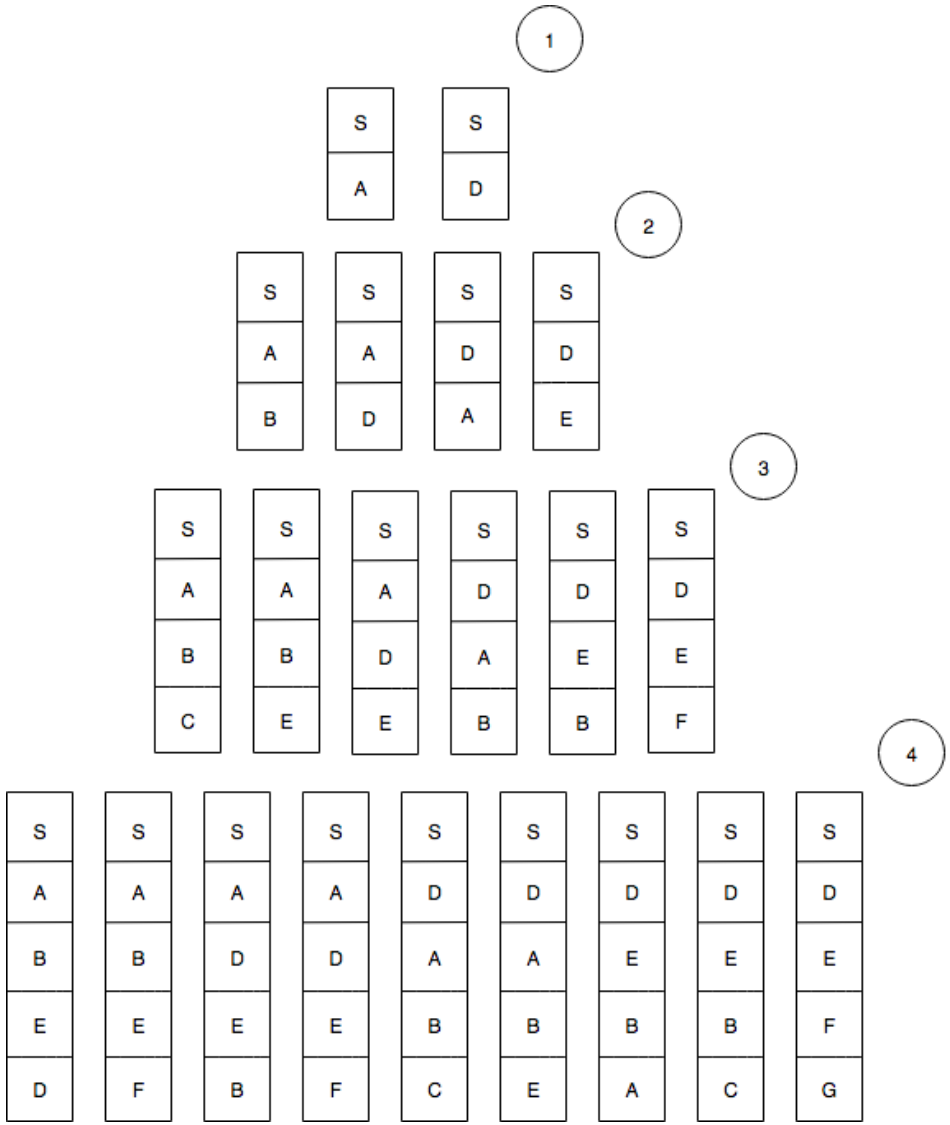




Búsqueda a Primera a lo Ancho (Breadth First Search)

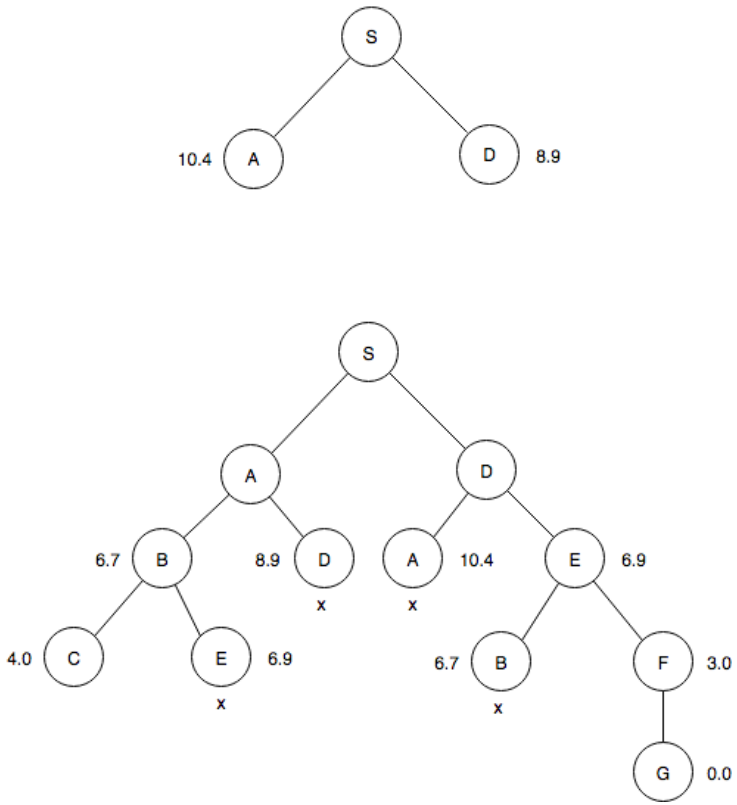


Búsqueda Primera a lo Ancho (Breadth First Search)



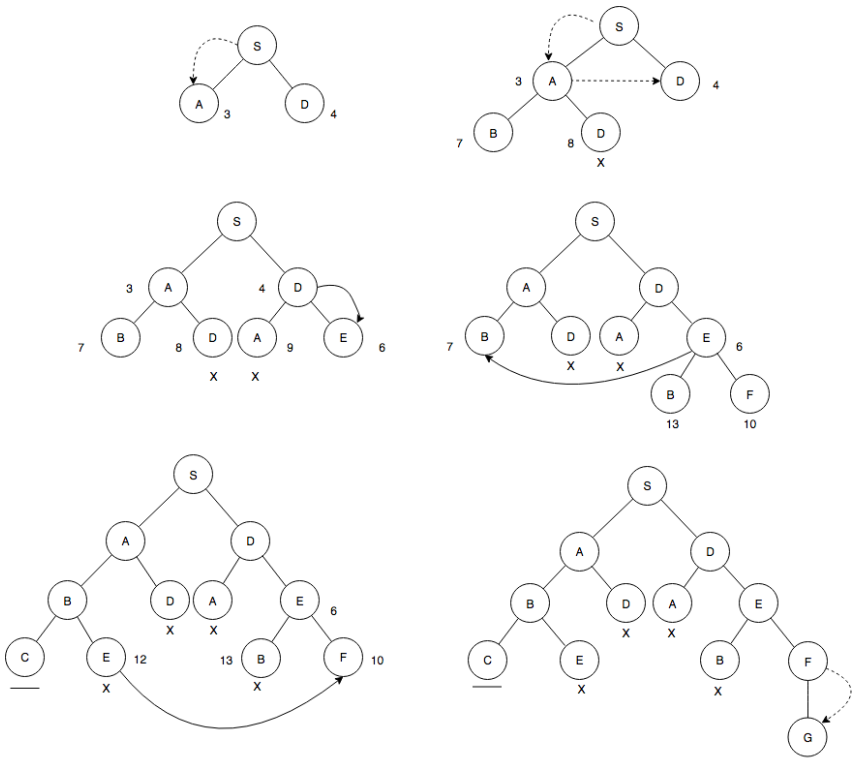
Búsqueda Dirigida (Beam Search)

Es igual a búsqueda primero a lo ancho , pero usa también una función heurística como en el metodo subiendo el monte.



Busqueda de Saltar y Límitar con Programación Dinámica

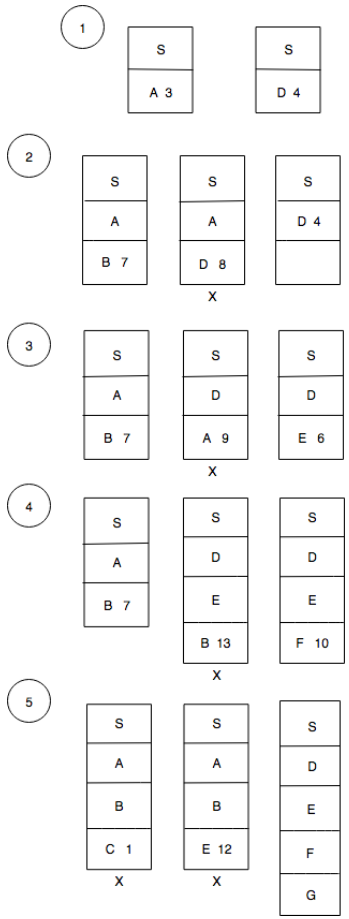
En este metodo se toman las distancias acumuladas al ir recorriendo los nodos y se van eliminando los caminos que llevan a los nodos con un mayor costo. La busqueda se va haciendo en los nodos con menor costo.



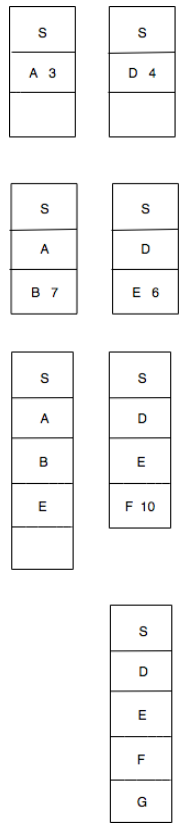
S - D - E - F - G

Busqueda de Saltar y Límitar con Programación Dinámica

Acomoda las colas tomando en cuenta la suma del costo acumulado hasta ahora, poniendo las rutas con el costo menor en el frente.



Busqueda de Saltar y Límitar con Programación Dinámica

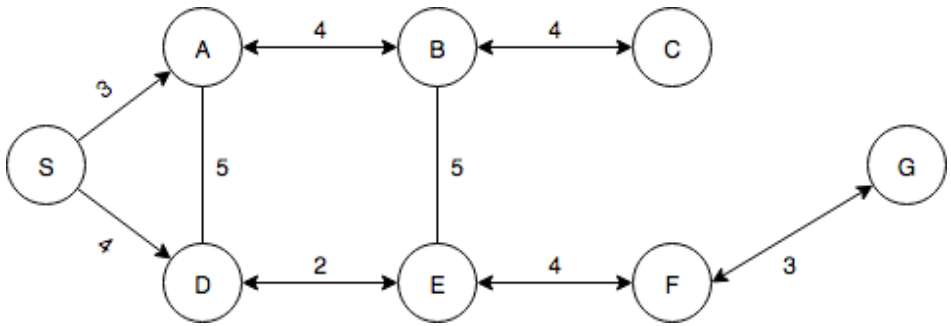


Algoritmo A*

La búsqueda de saltar y limitar usa las distancias acumuladas, A* usa una función heurística junto con el costo acumulado.

Algoritmo de Dijkstra

El algoritmo de Dijkstra funciona en etapas. En cada etapa selecciona el menor de los vértices etiquetados como desconocidos, y declara que el camino más corto del nodo **s** al nodo **v** desconocido. La parte final de cada etapa es actualizar los valores de la distancia acumulada d_j, de los vértices adyacentes de **v**, si $d_j = \infty$ entonces $d_w = d_v + C_v$, para el nodo adyacente w.



Algoritmo de Dijkstra

Se empieza con el nodo inicial S y se indica que es conocido. Se encuentra con que nodos esta conectado encontrandose los caminos más cortos a esos nodos.

v	*Conocido*	dv	Pv
S	1	0	0
A	0	3	S
D	0	4	S
B	0	∞	0
E	0	∞	0
C	0	∞	0
F	0	∞	0
G	0	∞	0

A pesar de que se puede ir de A a D no se sustituye ya que la distancia de S a D es menor que de A a D.

Algoritmo de Dijkstra

Una vez que se etiqueta un nodo como conocido, es decir que no hay otra conexión hacia el nodo con un costo menor, se encuentra con quien se conecta. Se busca siempre el nodo a desarrollar con menor costo y se le etiqueta como conocido. Se escoge A por tener un menor costo y se encuentra con quien se conecta.

v	*Conocido*	dv	Pv
S	1	0	0
A	1	3	S
D	0	4	S
B	0	7	A
E	0	∞	0
C	0	∞	0
F	0	∞	0
G	0	∞	0

Algoritmo de Dijkstra

Se escoge D por tener el costo menor y se expande

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	0	7	A
E	0	6	D
C	0	∞	0
F	0	∞	0
G	0	∞	0

Algoritmo de Dijkstra

Se escoge E por tener el costo menor y se expande

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	0	7	A
E	1	6	D
C	0	∞	0
F	0	10	E
G	0	∞	0

Algoritmo de Dijkstra

Se escoge B por tener el costo menor y se expande

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	1	7	A
E	1	6	D
C	0	11	B
F	0	10	E
G	0	∞	0

Algoritmo de Dijkstra

Se escoge F por tener el costo menor y se expande

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	1	7	A
E	1	6	D
C	0	11	B
F	1	10	E
G	0	13	F

Algoritmo de Dijkstra

Se escoge F por tener el costo menor y se expande

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	1	7	A
E	1	6	D
C	0	11	B
F	1	10	E
G	0	13	F

Algoritmo de Dijkstra

Se escoge C por tener el costo menor y se expande

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	1	7	A
E	1	6	D
C	1	11	B
F	1	10	E
G	0	13	F

Algoritmo de Dijkstra

Finalmente se escoge G

v	*Conocido*	dv	Pv
S	1	0	S
A	1	3	S
D	1	4	S
B	1	7	A
E	1	6	D
C	1	11	B
F	1	10	E
G	1	13	F

Con esta tabla se puede encontrar de como ir del nodo S a cualquier otro nodo. Para ir de S a G se hace un seguimiento hacía atrás empezando con G:

$$G,F,E,D,S$$