

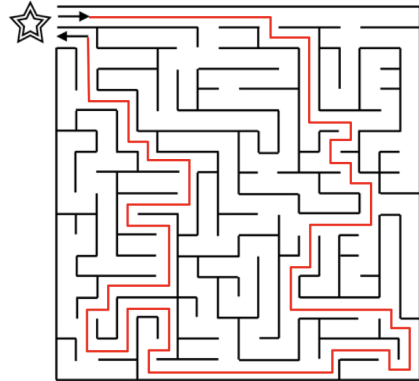
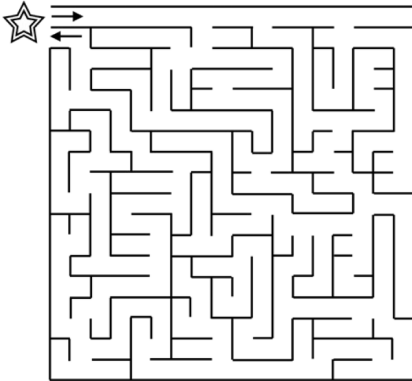
MÁS MÉTODOS DE BÚSQUEDA

ALAN REYES-FIGUEROA
INTELIGENCIA ARTIFICIAL

(AULA 06) 26.ENERO.2024

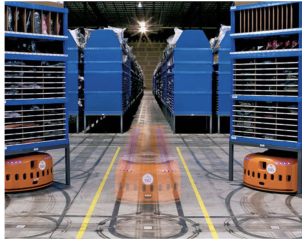
Problemas de Búsqueda

Aplicaciones: Diseño de rutas



Problemas de Búsqueda

Aplicaciones: Diseño de rutas



Commercial



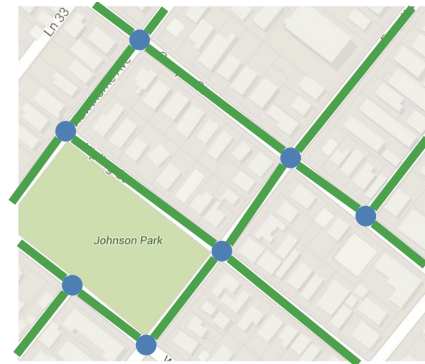
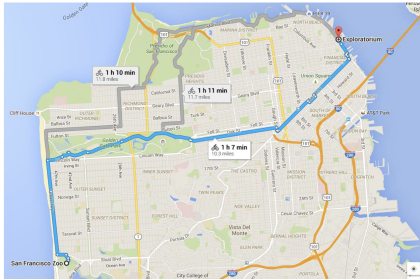
Search & Rescue



Domestic

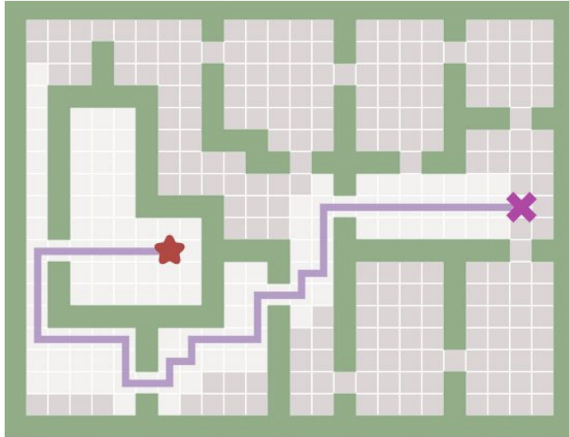
Problemas de Búsqueda

Aplicaciones: Diseño de rutas

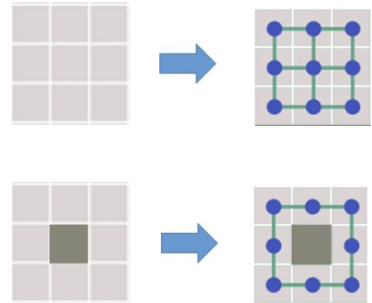
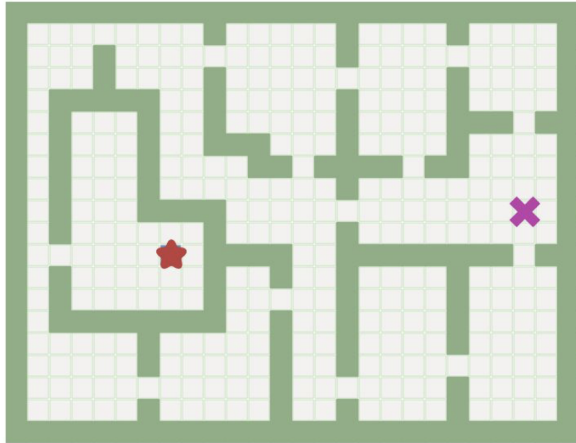


Problemas de Búsqueda

Aplicaciones: (*Shortest path*)

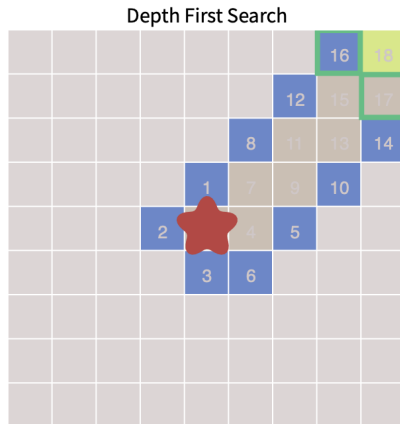
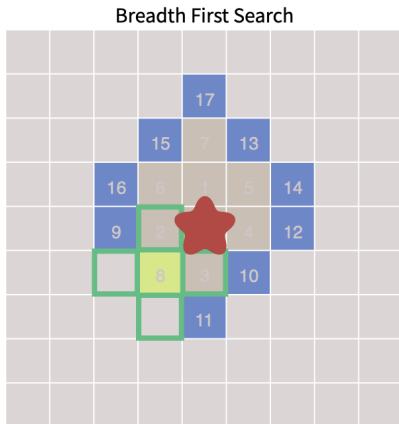


Problemas de Búsqueda



(a) Grid o malla, (b) Representación del grid como grafo.

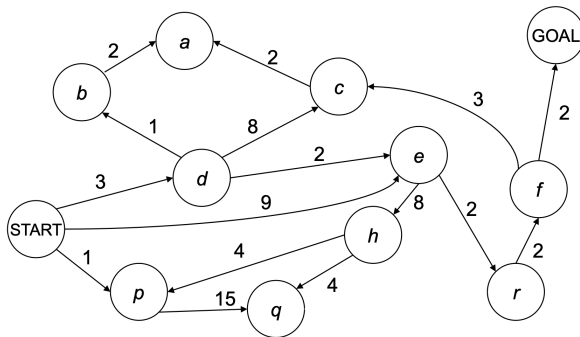
DFS vs. BFS



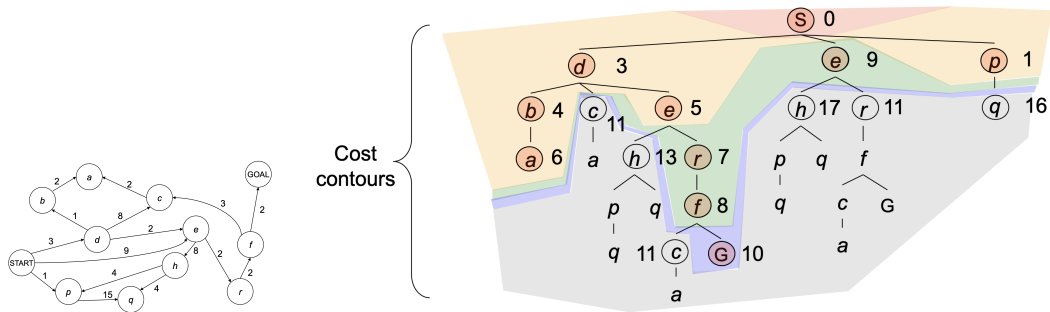
Contraste entre dos métodos de búsqueda: (a) BFS, (b) DFS.

Búsqueda sensible al Costo

BFS encuentra el camino más corto en términos de número de acciones. No encuentra el camino de menor costo. Cubrimos ahora un algoritmo similar que encuentra el camino de menor costo.



Búsqueda sensible al Costo



- Estrategia: expandir primero el nodo de menor costo. La franja o frontera usualmente se implementa como un *priority queue*.
- Prioridad: costo acumulado.

Búsqueda en Anchura (BFS)

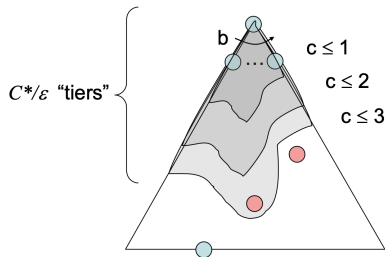
¿Cuáles nodos expande el BFS?

- Procesa todos los nodos cuyo costo es menor a la menor solución encontrada.
- Si esta solución es de costo C^* , y las aristas cuestan al menos ε , la **profundidad efectiva** es $O(C^*/\varepsilon)$.
- Requiere tiempo aproximado de $O(b^{C^*/\varepsilon})$.
- Requiere espacio de $O(b^{C^*/\varepsilon})$.

¿Es completo?

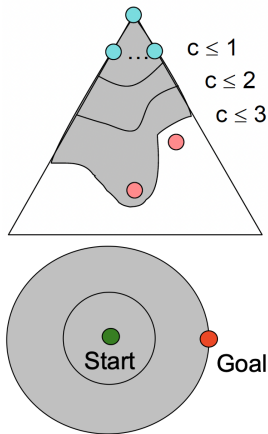
- Asumiendo que la mejor solución es de costo finito y el costo mínimo es > 0 (aristas de costo positivo), entonces sí.

¿Es optimal? Sí (se puede demostrar matemáticamente).



Búsqueda en Costo Uniforme (UCS)

- UCS explora todos los nodos, incrementando contornos de costos (curvas de nivel).
- Esto es bueno, en el sentido que UCS es completo y optimal.
- Lo malo: Explora opciones en todas las direcciones posibles.
- No posee información sobre la localización o estado objetivo..



Todos estos algoritmos de búsqueda son idénticos, excepto por las estrategias de exploración (franja o conjunto frontera).

- Conceptualmente, todos las fronteras son colas de prioridad (*priority queues*), es decir, colecciones de nodos con prioridades adjuntas.
- En la práctica, para DFS y BFS, se puede evitar la sobrecarga $\log(n)$ del *priority queue*, mediante el uso de una pila.

¿Cómo mejoramos esto?

Búsqueda Informada

Heurística: Una **heurística** es una función que estima qué tan cerca (o lejos) estamos del estado o configuración objetivo.

Son diseñadas específicamente para cada problema de búsqueda.

