# Tecnicas de Busqueda

Para poder aplicar esta técnica es esencial poder describir el problema como una búsqueda de configuraciones exitosas a partir de ciertas configuraciones iniciales, mediante la aplicación de reglas predefinidas de reconfiguración o avance

En general un problema P corresponde a un problema de búsqueda cuando se cuenta con:

* Una descripción precisa de que constituye el estado o configuración.
* Una descripción del estado de partida (estado inicial)
* Una descripción de aquellas configuraciones que se consideran exitosas (estados finales)
* Una descripción de cómo se puede avanzar, en un movimiento, desde un estado hacia otros estados sucesores.

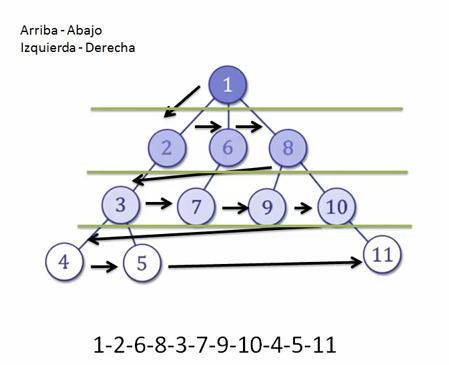
El problema es entonces encontrar, a partir del estado de partida, alguna configuración exitosa, usando las reglas de avance predefinidas.

**Estrategias de búsquedas**

**Breadth-first search:**  visita todos los nodos hasta encontrar un estado exitoso.

ventaja : Si existe un estado exitoso de profundidad finita, entonces será encontrado. Encuentra soluciones de profundidad mínima.

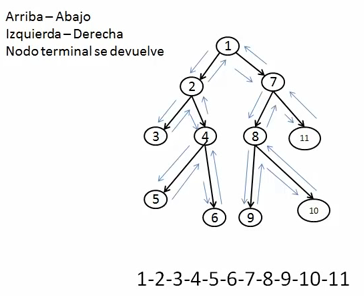
desventaja: Es necesario almacenar un conjunto de nodos exponencial con respecto al nivel que se está explorando.



**Depth-first search:**  visita los nodos en profundidad, considerando los hijos de un nodo antes que sus hermanos, hasta encontrar un estado exitoso.

ventajas: Solo es necesario almacenar un conjunto de nodos linealmente proporcional a la profundidad que se está explorando.

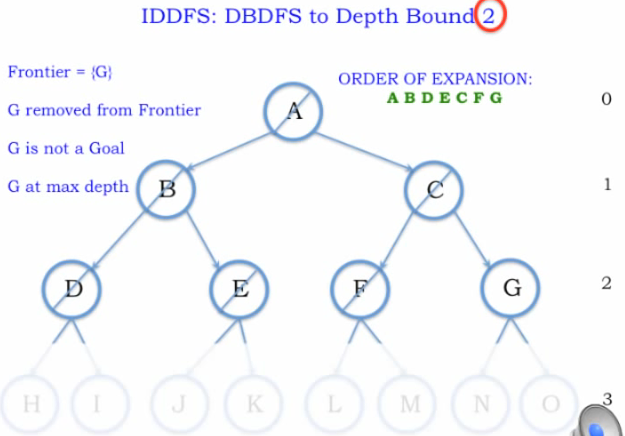
desventajas: Las soluciones encontradas no son necesariamente de profundidad mínima. Aun habiendo estados exitosos de profundidad finita, puede no encontrarlos ( puede perderse en ramas infinitas sin soluciones)



**Iterative Deepening:**

Consiste en realizar búsquedas depth-first search, cada una con un límite de profundidad incrementado en uno con respecto al anterior. Es decir:

* realizar un DFS hasta la profundidad 1
* si no se encontro una solucion, realizar DFS hasta la profundidad 2
* si no se encontro una solucion, realizar DFS hasta la profundidad 3
* …



Desventaja: Se vuelve a visitar múltiples veces un gran número de nodos.

Ventajas:

* Si existe un estado exitoso de profundidad finita, entonces será encontrado.
* Encuentra soluciones de profundidad mínima.
* Solo es necesario almacenar un conjunto de nodos linealmente proporcional a la profundidad que se está explorando.
* Las múltiples visitas a nodos no afecta la tasa de crecimiento asociada al tiempo de ejecución. (similar a depth-first search)

----------------------------------------------------------------------------------------------------------------------------

**Tecnicas de busqueda informada**

La información adicional requerida por algunas técnicas de mejora de algoritmos de visita es en general una estimación de la “Bondad” de un estado no final, que permite decidir que caminos seguir durante el recorrido del árbol de búsqueda.

Es un tipo de información extra muy simple que puede codificarse fácilmente como una función de valoración de estados.

*Técnica de la Escalada: (Hill Climbing)*

Esta tecnica utiliza una funcion de valoracion de Estados.

1- comienza con el estado inicial como estado actual.

2(a) - Si el estado actual es un exitoso, termina.

2(b)- Sino, obtener los hijos del estado actual, uno a la vez. Hasta encontrar uno con mayor valoracion que el estado actual o agotar todos sus hijos.

3(a) - Si no existen hijos con mayor valoracion que el estado actual, termina.

3(b) - Sino, volver al paso 2, con el primer hijo con mayor valoración que el actual.

Tecnica BEST-FIRST SEARCH:

Esta técnica es una variante de la técnica general de visita breadth-first search, que aprovecha la información provista por la función de valoración de estados.

1. comenzar con una cola ABIERTOS solo conteniendo al estado inicial.

2(a) Si abiertos esta vacia, terminar.

2(b) Sino, tomar al mejor estados de ABIERTOS como el estado actual.

3(a) si el estado actual es un estado exitoso, terminar exitosamente

3(b) Sino, obtener los hijos del estado actual y agregarlos a ABIERTOS

1. Quitar el estado actual de ABIERTOS
2. volver al paso 2.

Su performance depende fundamentalmente de la calidad de la función HEURISTICA.

puede no encontrar estados exitosos de profundidad finita.

**Si la función de valoración no es la adecuada, puede degenerar en una búsqueda depth-first, con un empeoramiento considerable en el espacio de almacenamiento utilizado.**

**Búsquedas en problemas con adversarios.**

Podemos reconocer algunas características propias de los problemas de búsqueda con adversarios:

* Existe un estado inicial bien definido
* A partir del estado inicial, se avanza mediante reglas de avance bien definidas, alternando movimientos con un adversario.
* El objetivo del adversario, al igual que el nuestro, es conseguir llegar a un estado exitoso.
* En general si el adversario tiene éxito, nosotros fracasamos.

A diferencia de los problemas de búsqueda convencionales, en problemas con adversarios el avance no depende de un solo “AGENTE”

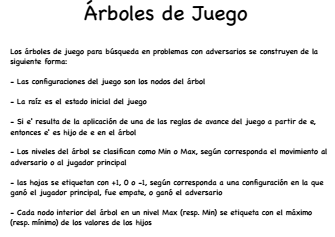
Tecnica MIN-MAX

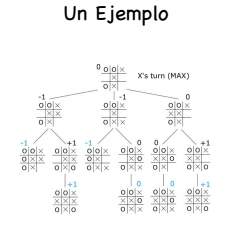
<http://joseluisbustos-meta4-peoplenet.blogspot.com.ar/2012/03/algoritmo-minimax.html>

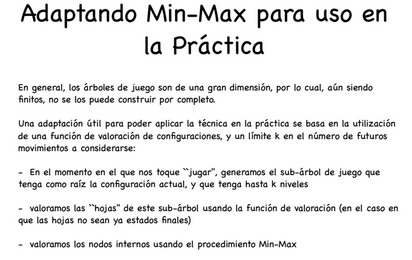
En juegos bipersonales el algoritmo más usado es el denominado Minimax. El procedimiento de búsqueda Minimax es una búsqueda en profundidad (DFS) de profundidad limitada. La idea consiste en comenzar en la posición actual y usar el generador de movimientos para generar las posibles posiciones sucesivas hasta un cierto límite de niveles. A continuación se aplica la función de evaluación estática a las posiciones obtenidas y se elige la mejor posición para el jugador correspondiente, llevando los valores un nivel hacia atrás para continuar la evaluación en todos los niveles anteriores.

Se supone una función de evaluación estática que devuelve valores elevados para indicar buenas situaciones y valores negativos para indicar buenas situaciones para el oponente. Visto de esta manera, la meta es maximizar el valor de la función estática de la siguiente posición de tablero.

El nombre del algoritmo deriva de considerar que, dada una función estática que devuelve valores en relación al jugador maximizarte, éste procura maximizar su valor mientras que su oponente procura minimizarlo. En un árbol de juego donde los valores de la función estática están en relación al jugador maximizarte, se maximiza y minimiza alternadamente de un nivel a otro.



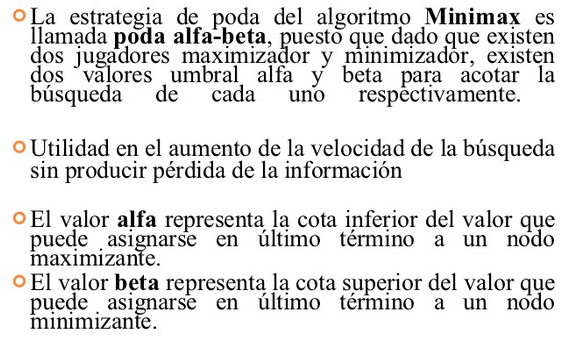




# 

# Poda Alfa-Beta: Una Mejora a Min-Max

El problema de la búsqueda Minimax es que el número de estados a explorar es exponencial al número de movimientos. Partiendo de este hecho, la técnica de poda alfa-beta trata de eliminar partes grandes del árbol, aplicándolo a un árbol Minimax estándar, de forma que se devuelva el mismo movimiento que devolvería este, gracias a que la poda de dichas ramas no influye en la decisión final.



### Ejemplo de poda alfa-beta