

## IDDFS (Iterative Deepening Depth-First Search)

### Funcionamento

- **Combinação de DFS e BFS:** O IDDFS utiliza a profundidade limitada da DFS, mas de forma iterativa, aumentando gradualmente a profundidade máxima permitida.
- **Chamadas Iterativas:** Realiza múltiplas chamadas de DFS com diferentes profundidades, começando do zero e incrementando até encontrar o nó alvo.
- **Limitação de Profundidade:** Cada chamada de DFS é limitada pela profundidade máxima da iteração atual.

### Complexidade de Tempo

- **Complexidade Assintótica:** Igual ao DFS e BFS, mas com um fator constante mais alto devido às múltiplas visitas aos níveis superiores.

### Vantagens

- **Eficiência de Espaço:** Usa espaço proporcional à profundidade máxima do nó alvo.
- **Busca Completa:** Completo para gráficos infinitos.
- **Rapidez para Nós Próximos à Raiz:** Encontra nós próximos à raiz rapidamente.

### Desvantagens

- **Maior Custo Constante:** Mais lento que DFS e BFS devido às visitas repetidas aos níveis superiores.
- **Repetição de Trabalho:** Ineficiência devido à visita repetida aos nós superiores.

### Aplicações

- Adequado para busca em árvores infinitas ou grandes onde a profundidade da solução é desconhecida.

## Busca Bidirecional

### Funcionamento

- **Dois Pontos de Partida:** Inicia a busca simultaneamente a partir do nó inicial e do nó alvo.
- **Encontrar-se no Meio:** As duas buscas progridem até se encontrarem, reduzindo o espaço de busca total.

### Complexidade de Tempo

- **Redução de Complexidade:** Cada busca percorre aproximadamente metade do caminho, resultando em uma redução significativa na complexidade, geralmente

de  $O(bd)O(b^d)O(bd)$  para  $O(bd/2)O(b^{\lceil d/2 \rceil})O(bd/2)$ , onde  $b$  é o fator de ramificação e  $d$  é a profundidade da solução.

### Vantagens

- **Eficiência de Tempo:** Muito mais rápida que a busca unidirecional em gráficos com grande fator de ramificação.
- **Menos Exploração:** Explora menos nós em comparação com a busca tradicional, devido à divisão do espaço de busca.

### Desvantagens

- **Sincronização:** Necessidade de gerenciar e sincronizar duas buscas simultâneas.
- **Conhecimento do Alvo:** Requer conhecimento do nó alvo desde o início, o que nem sempre é possível.
- **Armazenamento de Fronteiras:** Pode exigir mais espaço para armazenar as fronteiras de ambas as buscas.

### Aplicações

- Adequada para problemas onde a posição do nó alvo é conhecida e a busca em gráficos grandes é necessária, como em navegação de robôs e jogos de labirinto.