Study Guide: Uniform-Cost Search, Depth-First Search, and Variants

I. Uniform-Cost Search

* Definition: A variant of Breadth-First Search, where the cost of the path is considered when choosing the next node to expand.
* Characteristics:
  + Implementation: fringe = queue ordered by cost, lowest first
  + Equivalent to Breadth-First Search if step costs are all equal
* Applications:
  + Finding in graph search
* Example: Calculating the minimum cost to transfer 100MB of data in 24 hours, given a transfer speed of 100MB/sec.
  + 8640GB = 100MB/sec × 24 hours
* Diagram: (ASCII art not possible for this example)

II. Depth-First Search

* Definition: A search algorithm that picks the deepest unexpanded node.
* Characteristics:
  + Implementation: fringe = LIFO queue, put successors at front
  + Not complete in infinite-depth spaces or spaces with loops
* Applications:
  + Graph search
* Example: Page 21 of the provided context
* Diagram:

```
    A
   / \
  B   C
```

```
  /\  \
 D  E  F
```

In this example, the deepest unexpanded node is chosen, resulting in a Depth-First Search.

III. Depth-Limited Search

* Definition: A variant of Depth-First Search with a depth limit.
* Characteristics:
  + Implementation: recursive function with a depth limit
  + Not complete in infinite-depth spaces
* Applications:
  + Graph search with a limited depth
* Example: Page 22 of the provided context
* Diagram: (ASCII art not possible for this example)

IV. Iterative Deepening Search

* Definition: A search algorithm that combines Depth-First Search with increasing depth limits.
* Characteristics:
  + Implementation: iterative deepening of the depth limit
  + Complete and optimal if step cost = 1
* Applications:
  + Graph search with optimal solutions
* Example: Page 23 of the provided context
* Diagram: (ASCII art not possible for this example)

Summary of Key Points:

* Uniform-Cost Search: complete, optimal, and time complexity O(b^d+1), space complexity O(b^d+1)
* Depth-First Search: not complete, not optimal, time complexity O(b^m), space complexity O(b^m)
* Depth-Limited Search: not complete, not optimal, time complexity O(b^l), space complexity O(b^l)
* Iterative Deepening Search: complete, optimal, time complexity O(b^d), space complexity O(b^d)

Flashcards:

1. Q: What is Uniform-Cost Search?
A: A variant of Breadth-First Search that considers path cost when choosing the next node to expand.

2. Q: What is Depth-First Search?
A: A search algorithm that picks the deepest unexpanded node.

3. Q: What is Depth-Limited Search?
A: A variant of Depth-First Search with a depth limit.

4. Q: What is Iterative Deepening Search?
A: A search algorithm that combines Depth-First Search with increasing depth limits.

5. Q: What is the time complexity of Uniform-Cost Search?
A: O(b^d+1)