Lecture 6: Informed Search (Heuristics, Greedy Best-First Search, A*)

1. Recap from Last Lecture

- We studied **Uninformed Search (BFS & DFS)**.\
- Trade-offs: BFS = complete & optimal but memory-heavy; DFS = memory-light but incomplete & non-optimal.\
- Motivation: Can we guide the search **more intelligently**?

2. Informed Search

- Uses **heuristic knowledge** to guide search toward the goal.\
- A **heuristic (h(n))** is an estimate of the cost to reach the goal from node *n*.\
- Example: In a map navigation problem, heuristic = straight-line distance.

3. Heuristic Function

- **Definition:** Function *h(n)* that estimates the cheapest cost from node *n* to the goal.\
- **Admissible Heuristic:** Never overestimates actual cost.\
- **Consistent Heuristic:** For every node *n* and successor *n'*:
  [h(n) `\leq` {=tex}c(n, n') + h(n')]

Example Heuristics:

- Manhattan distance (for grids).\
- Euclidean distance.\
- Number of misplaced tiles (for 8-puzzle).

4. Greedy Best-First Search (GBFS)

- **Idea:** Always expand the node with the lowest *h(n)* (closest to goal estimate).\
- **Pros:** Fast, often finds a solution quickly.\
- **Cons:** Not complete, not optimal --- can get stuck in loops or dead ends.\
- **Data Structure:** Priority Queue ordered by *h(n)*.

5. A* Search

- **Idea:** Combines actual cost + heuristic:
  [f(n) = g(n) + h(n)] where:
  - *g(n)* = cost so far (from start to n)\
  - *h(n)* = estimated cost to goal\
- **Algorithm:** Expands node with smallest *f(n)*.\

- **Properties:**
  - **Complete** (if step cost ≥ ε > 0).\
  - **Optimal** (if *h(n)* is admissible).\
  - Widely used in pathfinding (games, robotics, logistics).

---

6. BFS, Greedy, and A* Comparison

Algorithm Uses g(n) Uses h(n) Complete? Optimal? Memory

BFS ✔ ✖ ✔ ✔ (if High uniform
costs)

GBFS ✖ ✔ ✖ ✖ Low

A* ✔ ✔ ✔ ✔ (if Moderate admissible h)

---

7. Example: Route Planning

- Task: Find shortest route from **Arad** → **Bucharest** (Romania map from AIMA).\
- **Heuristic:** Straight-line distance to Bucharest.\
- BFS explores too many nodes.\
- GBFS follows the "closest" city but risks dead ends.\
- A* balances both cost so far and estimated cost → finds the optimal path.

---

8. Key Takeaways

- Heuristics make search efficient.\
- GBFS is fast but unreliable.\
- A* is the gold standard: complete + optimal (with admissible heuristic).

---

9. Reading & Exercises

- **Reading:** AIMA, Ch. 3 (Informed Search section).\
- **Exercise:**
  - Implement A* search for an 8-puzzle or maze problem.\
  - Compare number of nodes expanded by BFS vs GBFS vs A*.\
- **Discussion Question:** *Why is admissibility of heuristics crucial for A*?*