

INSTITUTION OF TECHNOLOGY SCHOOL OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

Course Title: Fundamentals of Artificial Intelligence

Course Title: Real Time And Embedded System

Course code: SEng4033

Group Assignment

GROUP ONE (1)

Name ID

1. Daniel wasihun --------------------------------------------------- WDU145817
2. Mikiyas abera ----------------------------------------------------- WDU146855

1. Esayas Abiyu --------------------------------------------------------WDU148959
2. Mebt Andualem ----------------------------------------------------WDU148917
3. Ahmed Tlahun ------------------------------------------------------WDU146519
4. Haylemariam Semere-----------------------------------------------WDU1306338

**Submitted to: Instructor Teame S.**

**May 16 ,2017 E.C**

**Woldia, Ethiopia**

Contents

[Step 1: Understand the Problem and Context 3](#_Toc198897552)

[Step 2: Analyze Each Person’s Situation 3](#_Toc198897553)

[Person 1: Knows Nothing, Wants to Hide in Another City for 3 Months 3](#_Toc198897554)

[Person 2: Can Communicate, Gets Estimated Distances (Heuristic Values) 5](#_Toc198897555)

[Person 3: Has a City Map with Actual Distances 7](#_Toc198897556)

[Person 4: Can Communicate (Heuristic Values) and Has a City Map (Actual Distances) 9](#_Toc198897557)

[Step 3: Summarize the Strategies in a Table 11](#_Toc198897558)

# Step 1: Understand the Problem and Context

* **Goal**: Each person (except the first) starts from Big City and aims to reach Benin. The first person wants to hide in another city for 3 months.
* **Information Available**:
  + **Person 1**: Knows nothing, goal is to hide in any city other than the starting city.
  + **Person 2**: Can communicate to get estimated distances (heuristic values) to other cities.
  + **Person 3**: Has a city map with actual distances from the current city to other cities.
  + **Person 4**: Can communicate (gets heuristic values) and has a city map (actual distances).
* **Map Details**: The map includes cities with heuristic values (estimated distances to Benin, shown in circles) and actual values (real distances between cities). This is typical in AI search problems, where heuristics guide informed searches, and actual distances determine path costs.
* **Task**: Select the best search strategy for each person based on their information, ensuring alignment with AI search principles (e.g., uninformed vs. informed search). The strategies should be reasonable, considering completeness, optimality, and efficiency.

# Step 2: Analyze Each Person’s Situation

## Person 1: Knows Nothing, Wants to Hide in Another City for 3 Months

* **Information**: No map, no distances, no communication ability. The goal is to “hide in another city for 3 months,” implying reaching any city other than the starting city and staying there.
* **Starting Point**: Not explicitly Big City, but assumed to be Big City for consistency with others, as the problem doesn’t specify otherwise.
* **Goal**: Reach any city other than Big City (e.g., Benghazi, Kinshasa).
* **Challenges**: No knowledge of the map, distances, or connections, ruling out informed search strategies (e.g., A\*, Greedy Best-First Search) that require heuristics or actual costs.
* **Possible Strategies**:
  + **Breadth-First Search (BFS)**: Explores nodes level by level, ensuring the shallowest path (fewest steps) to any city, ideal for an uninformed search.
  + **Depth-First Search (DFS)**: Explores deeply along one path, but may take longer paths and is not optimal for step counts.
  + Since the goal is to hide quickly, BFS is preferred for finding the closest city (in steps).
* **Best Strategy**: Breadth-First Search (BFS)
* **Reasoning**:
  + BFS expands nodes in a FIFO manner, exploring all nodes at depth *d* before depth *d+1*, ensuring the shallowest path to another city. This is ideal for Person 1, who needs to reach any non-Big City quickly without map knowledge.
  + A visited set prevents revisiting Big City or looping, addressing potential cycles in the graph.
  + BFS is complete and optimal for unit step costs (assuming each move is 1 step for simplicity), though actual distances are used for final path cost.
* **Limitations**: High space complexity (O(b^{d+1})), but the graph is small (likely depth 1 for neighbors), making BFS practical.
* **Assumptions**:
  + Start: Big City.
  + Goal: Any city other than Big City (e.g., Benghazi).
  + Node Format: City name (no costs, as no distance info is available).
  + Priority: FIFO queue, alphabetical neighbor ordering.
* **Implementation of Breadth-First Search (BFS)**:
  + **Priority**: FIFO queue, expanding nodes at depth *d* before *d+1*, alphabetical neighbor ordering.
  + **Node Format**: City name, no cost (Person 1 has no distance info).
  + **NodeList**: Unexpanded nodes in the queue without costs.
  + **Execution**:

|  |  |
| --- | --- |
| Expanded Node | NodeList |
| Big City | {Big City} |
| Benghazi | {Giza, Kinshasa, Lagos, Petra, Varanasi} |

* + **Path**: Big City → Benghazi
  + **Cost**: 100 km (map distance for completeness)
  + **Number of Expanded Nodes**: 2 (Big City, Benghazi)
  + **Explanation**:
    - **Step 1**: Expand Big City (depth 0). Neighbors: Benghazi, Giza, Kinshasa, Lagos, Petra, Varanasi (alphabetical). Enqueue all. Goal check: Big City fails (is Big City). Visited: {Big City}. NodeList: {Benghazi, Giza, Kinshasa, Lagos, Petra, Varanasi}.
    - **Step 2**: Expand Benghazi (depth 1, first in queue). Neighbors: Big City (visited, skip), Laranca, Luxor. Goal check: Benghazi ≠ Big City, goal reached. Stop. Visited: {Big City, Benghazi}. NodeList: {Giza, Kinshasa, Lagos, Petra, Varanasi}.
    - **Logic**: BFS stops at the first goal node (Benghazi), as the goal is “any city other than Big City.” The actual cost (100 km) is computed from the map for consistency.
    - **Correction**: Unlike "AIPartOne.docx" (which continued to Kinshasa), BFS stops at Benghazi, the first valid goal, avoiding unnecessary expansions.

## Person 2: Can Communicate, Gets Estimated Distances (Heuristic Values)

* **Information**: Can ask people for estimated distances to Benin (heuristic values, e.g., straight-line distance), but no actual distances or map.
* **Starting Point**: Big City
* **Goal**: Reach Benin
* **Challenges**: Only heuristic values (h(n)) are available, ruling out strategies requiring actual costs (e.g., Uniform Cost Search, A\*).
* **Possible Strategies**:
  + **Greedy Best-First Search**: Prioritizes nodes with the lowest h(n), using heuristics to guide toward Benin.
  + *A Search*\*: Requires actual costs (g(n)), which Person 2 lacks, so it’s not applicable.
* **Best Strategy**: Greedy Best-First Search
* **Reasoning**:
  + Greedy Best-First Search uses h(n) to select the node that appears closest to Benin, fitting Person 2’s ability to get estimated distances.
  + A visited set prevents loops, ensuring progress.
  + It’s efficient but not guaranteed to find the optimal path, as it ignores actual costs.
* **Limitations**: May choose suboptimal paths if heuristics are misleading.
* **Assumptions**:
  + Heuristic values from "AIPartOne.docx" (e.g., h(Big City)=500, h(Giza)=200, h(Benin)=0).
  + Map connections used for path construction.
* **Implementation of Greedy Best-First Search**:
  + **Priority**: Lowest h(n).
  + **Node Format**: City name, cost is h(n).
  + **NodeList**: Cities with h(n) values.
  + **Execution**:

|  |  |
| --- | --- |
| Expanded Node | NodeList |
| Big City (500) | {Big City:500} |
| Giza (200) | {Giza:200, Lagos:400, Kinshasa:400, Benghazi:600, Petra:600, Varanasi:1000} |
| Benin (0) | {Benin:0, Lagos:400, Kinshasa:400, Ife:400, Big City:500, Benghazi:600, Petra:600, Varanasi:1000} |

* + **Path**: Big City → Giza → Benin
  + **Cost**: 330 + 90 = 420 km
  + **Nodes Expanded**: 3 (Big City, Giza, Benin)
  + **Explanation**:
    - **Step 1**: Expand Big City (h=500). Neighbors: Giza (h=200), Lagos (h=400), Kinshasa (h=400), Benghazi (h=600), Petra (h=600), Varanasi (h=1000). NodeList: {Giza:200, Lagos:400, Kinshasa:400, Benghazi:600, Petra:600, Varanasi:1000}.
    - **Step 2**: Expand Giza (h=200). Neighbors: Big City (h=500), Kinshasa (h=400), Ife (h=400), Benin (h=0). NodeList: {Benin:0, Lagos:400, Kinshasa:400, Ife:400, Big City:500, Benghazi:600, Petra:600, Varanasi:1000}.
    - **Step 3**: Expand Benin (h=0). Goal reached.
    - **Note**: The path is suboptimal (420 km vs. optimal 328 km), as Greedy prioritizes heuristics over actual costs.

## Person 3: Has a City Map with Actual Distances

* **Information**: Has a map with actual distances between cities (g(n)), but no heuristic values.
* **Starting Point**: Big City
* **Goal**: Reach Benin
* **Challenges**: No heuristics, so informed searches (Greedy, A\*) are not feasible. Actual distances enable cost-based search.
* **Possible Strategies**:
  + **Breadth-First Search (BFS)**: Optimal for equal step costs, but the map has varying distances.
  + **Uniform Cost Search (UCS)**: Expands nodes with the lowest g(n), ideal for actual distances.
* **Best Strategy**: Uniform Cost Search (UCS)
* **Reasoning**:
  + UCS guarantees the least-cost path by expanding the node with the lowest path cost (g(n)), fitting Person 3’s map-based information.
  + A visited set prevents re-expansion of cities.
  + Unlike BFS, UCS handles varying distances, ensuring optimality.
* **Limitations**: High time/space complexity (O(b^{d+1})) for large graphs, but optimal for actual costs.
* **Assumptions**:
  + Actual distances from "AIPartOne.docx" (e.g., Big City → Lagos = 98 km, Lagos → Addis Ababa = 120 km).
* **Implementation of Uniform Cost Search (UCS)**:
  + **Priority**: Lowest g(n).
  + **Node Format**: City, g(n) (actual path cost).
  + **NodeList**: Cities with g(n) values.
  + **Execution**:

|  |  |
| --- | --- |
| Expanded Node | NodeList |
| Big City (0) | {Big City:0} |
| Kinshasa (90) | {Kinshasa:90, Lagos:98, Benghazi:100, Petra:140, Giza:330, Varanasi:670} |
| Lagos (98) | {Lagos:98, Benghazi:100, Petra:140, Addis Ababa:218, Giza:410, Cairo:510, Varanasi:670} |
| Benghazi (100) | {Benghazi:100, Petra:140, Laranca:160, Addis Ababa:218, Giza:410, Cairo:510, Luxor:620, Varanasi:670} |
| Petra (140) | {Petra:140, Laranca:160, Addis Ababa:218, Cadiz:340, Luxor:350, Giza:410, Cairo:510, Luxor:620, Varanasi:670} |
| Laranca (160) | {Laranca:160, Addis Ababa:218, Cadiz:340, Luxor:350, Giza:410, Cairo:510, Luxor:620, Varanasi:670} |
| Addis Ababa (218) | {Addis Ababa:218, Cadiz:340, Luxor:350, Giza:410, Cairo:510, Luxor:620, Varanasi:670} |
| Benin (328) | {Benin:328, Cadiz:340, Luxor:350, Giza:410, Cairo:510, Luxor:620, Varanasi:670} |

* + **Path**: Big City → Lagos → Addis Ababa → Benin
  + **Cost**: 98 + 120 + 110 = 328 km
  + **Nodes Expanded**: 8
  + **Explanation**:
    - **Step 1**: Expand Big City (g=0). Neighbors: Kinshasa (90), Lagos (98), Benghazi (100), Petra (140), Giza (330), Varanasi (670).
    - **Step 2**: Expand Kinshasa (g=90). Neighbors: Cairo (510), Giza (410, updates Giza:330).
    - **Step 3**: Expand Lagos (g=98). Neighbors: Addis Ababa (218).
    - **Step 4**: Expand Benghazi (g=100). Neighbors: Laranca (160), Luxor (620).
    - **Step 5**: Expand Petra (g=140). Neighbors: Cadiz (340), Luxor (350).
    - **Step 6**: Expand Laranca (g=160). No new unvisited neighbors.
    - **Step 7**: Expand Addis Ababa (g=218). Neighbors: Benin (328).
    - **Step 8**: Expand Benin (g=328). Goal reached.
    - **Note**: Optimal path (328 km), as UCS minimizes g(n).

## Person 4: Can Communicate (Heuristic Values) and Has a City Map (Actual Distances)

* **Information**: Has both heuristic values (estimated distances to Benin) and actual distances (map).
* **Starting Point**: Big City
* **Goal**: Reach Benin
* **Challenges**: Has the most information, enabling advanced search strategies.
* **Possible Strategies**:
  + **Greedy Best-First Search**: Uses h(n), but ignores actual costs, so not optimal.
  + *A Search*\*: Combines g(n) and h(n) for optimal path finding.
* **Best Strategy**: A\* Search
* **Reasoning**:
  + A\* Search uses f(n) = g(n) + h(n) to expand nodes with the lowest estimated total cost, ensuring optimality if the heuristic is admissible (h(n) ≤ actual cost to Benin).
  + Person 4’s access to both actual and heuristic distances makes A\* the best choice, balancing efficiency and optimality.
* **Limitations**: Higher memory usage than Greedy, but optimal with admissible heuristics.
* **Assumptions**:
  + Heuristic and actual distances from "AIPartOne.docx" (e.g., h(Big City)=500, Big City → Lagos = 98 km).
  + Heuristic is admissible (e.g., h(Big City)=500 ≥ 328, h(Benin)=0).
* *Implementation of A Search*\*:
  + **Priority**: Lowest f(n) = g(n) + h(n).
  + **Node Format**: City, f(n).
  + **NodeList**: Cities with f(n) values.
  + **Execution**:

|  |  |
| --- | --- |
| Expanded Node | NodeList |
| Big City (500) | {Big City:500} |
| Kinshasa (490) | {Kinshasa:490, Lagos:498, Giza:530, Benghazi:700, Petra:740, Varanasi:1670} |
| Lagos (498) | {Lagos:498, Giza:530, Addis Ababa:418, Benghazi:700, Petra:740, Cairo:1110, Varanasi:1670} |
| Addis Ababa (418) | {Addis Ababa:418, Giza:530, Benghazi:700, Petra:740, Cairo:1110, Varanasi:1670} |
| Benin (328) | {Benin:328, Giza:530, Benghazi:700, Petra:740, Cairo:1110, Varanasi:1670} |

* + **Path**: Big City → Lagos → Addis Ababa → Benin
  + **Cost**: 98 + 120 + 110 = 328 km
  + **Nodes Expanded**: 5
  + **Explanation**:
    - **Step 1**: Expand Big City (g=0, h=500, f=500). Neighbors: Kinshasa (f=90+400=490), Lagos (f=98+400=498), Giza (f=330+200=530), Benghazi (f=100+600=700), Petra (f=140+600=740), Varanasi (f=670+1000=1670).
    - **Step 2**: Expand Kinshasa (f=490). Neighbors: Cairo (f=510+600=1110), Giza (f=410+200=610, updates Giza:530).
    - **Step 3**: Expand Lagos (f=498). Neighbors: Addis Ababa (f=218+200=418).
    - **Step 4**: Expand Addis Ababa (f=418). Neighbors: Benin (f=328+0=328).
    - **Step 5**: Expand Benin (f=328). Goal reached.
    - **Note**: Optimal path (328 km, matches UCS), with fewer expansions (5 vs. 8) due to heuristic guidance. Heuristic is admissible.

# Step 3: Summarize the Strategies in a Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Person | Information Available | Starting Point | Goal | Best Search Strategy | Reasoning | Limitations |
| Person 1 | Knows nothing | Big City | Hide in another city | Breadth-First Search (BFS) | No map; BFS finds shallowest path to any city, visited set avoids loops. | High space complexity, not optimal for actual costs. |
| Person 2 | Heuristic values (estimated distances) | Big City | Reach Benin | Greedy Best-First Search | Uses h(n) to guide toward Benin; efficient but suboptimal. | Not optimal, may follow misleading heuristics. |
| Person 3 | Actual distances (city map) | Big City | Reach Benin | Uniform Cost Search (UCS) | Uses g(n) for least-cost path; optimal with actual distances. | High time/space complexity for large graphs. |
| Person 4 | Heuristic and actual distances | Big City | Reach Benin | A\* Search | Combines g(n) and h(n) for optimal path; efficient with admissible heuristic. | High memory usage, requires admissible heuristic. |

**Final Notes**

* **Person 1**: BFS is chosen over DFS (from "AIPartOne.docx") for its guarantee of the shallowest path, ideal for quickly hiding. Stops at Benghazi (100 km), correcting unnecessary expansions in the original document.
* **Person 2**: Greedy Best-First Search yields a suboptimal path (420 km), as expected, due to heuristic-only guidance.
* **Person 3 and 4**: Both UCS and A\* find the optimal path (328 km). A\* is more efficient (5 expansions vs. 8) due to heuristics, making it ideal for Person 4.
* **Map**: Cities are mapped as A (Big City), F (Lagos), N (Addis Ababa), M (Benin), etc., based on "AIPartOne.docx" connections and distances.
* **Heuristic**: Assumed admissible for A\* (e.g., h(Big City)=500 ≥ 328, h(Benin)=0), ensuring optimality.
* This response mirrors the structure of "AI\_GROUP\_WORK\_PART\_ONE\_GROUP\_8.docx" with clear steps, detailed implementations, and a summary table, while integrating accurate implementations from "AIPartOne.docx" and correcting errors (e.g., BFS stopping at Benghazi).