**Student:**  Bunge, Shea

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Criteria | HD (High Distinction) | DN (Distinction) | CR (Credit) | PP (Pass) | NN (Fail) |
|  | **You have:** | **You have:** | **You have:** | **You have:** | **You have:** |
| 1. Implement code to construct an adjacency list representation of a graph   ***Weighting 30%*** | * Correctly used the provided functions to generate an adjacency list from the DEM * Used correct data structures * Written code that is clear and easy to follow * Graph memory is freed when program terminates | * Correctly used the provided functions to generate an adjacency list from the DEM * Used correct data structures * Written code that is clear and easy to follow | * Correctly used the provided functions to generate an adjacency list from the DEM * Used correct data structures | * Used the provided functions to generate an adjacency list with some errors * Used correct data structures | * No serious attempt or incorrect data structures |
| **Comment:** Excellent **HD+** | | | | | |
| 1. Implement Dijkstra’s algorithm to find shortest paths   ***Weighting 35%*** | * Correctly implemented function to find path distances using Dijkstra * Correctly implemented function to construct shortest paths * Used correct data structures * Written code that is clear and easy to follow * Distance and path memory is freed when no longer needed | * Correctly implemented function to find path distances * Correctly implemented function to return shortest paths * Written code that is clear and easy to follow * Used correct data structures | * Correctly implemented function to find path distances * Correctly implemented function to return shortest paths * Used correct data structures | * Implemented functions with some errors * Correct data structures used | * No serious attempt or incorrect data structures |
| **Comment:** Ditton. **HD+** | | | | | |
| 1. Implement Floyd-Warshall algorithm to find shortest paths   ***Weighting 35%*** | * Correctly implemented function to find path distances using Floyd-Warshall * Correctly implemented function to return shortest paths * Used correct data structures * Written code that is clear and easy to follow * Distance and path memory is freed when no longer needed | * Correctly implemented function to find path distances * Correctly implemented function to return shortest paths * Written code that is clear and easy to follow * Used correct data structures | * Correctly implemented function to find path distances * Correctly implemented function to return shortest paths * Used correct data structures | * Implemented functions with some errors * Correct data structures used | * No serious attempt or incorrect data structures |
| **Comment:** And again. **HD+** | | | | | |

**HD+ (100)** Excellent work and very well presented code. I would have to be really picky and pedantic to find faults. Well done.