**Little Heath Sixth Form**

**Mathematics** Personal Learning Checklist

**Student Name: ……………………….…………………………………..………**

|  |  |
| --- | --- |
| **Unit Name:**  **Mathematics (Decision 2)** | **Unit Code:**  **MD02** |
| *Minimum Target Grade:* | *Aspirational Target Grade:* |

*KEY:* ***Red =*** *with difficulty* ***Amber*** *= not sure* ***Green*** *= yes*

|  |  |  |  |
| --- | --- | --- | --- |
| **D1 Re-Cap (Skills and Knowledge)** | **Red** | **Amber** | **Green** |
| * Understand the concept of an algorithm |  |  |  |
| * Understand and be familiar with Linear Programming |  |  |  |
| * Draw a bipartite graph |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Allocation-The Hungarian Algorithm** | **Red** | **Amber** | **Green** | **To address this before the exam I will:-** |
| * Understand how to formulate an allocation problem in matrix form |  |  |  |  |
| * Formulate a square cost matrix |  |  |  |  |
| * Work out an opportunity cost matrix by row and column reduction |  |  |  |  |
| * Be able to apply the line test successfully |  |  |  |  |
| * Be able to augment an opportunity cost matrix |  |  |  |  |
| * Successfully establish a minimal matching |  |  |  |  |
| * Be able to adapt the algorithm to find a maximal matching |  |  |  |  |
| * Be able to adapt the algorithm to use on a non-square matrix |  |  |  |  |
| **Dynamic Programming** |  |  |  |  |
| * Understand and use Bellman’s principle |  |  |  |  |
| Understand and use stages and states |  |  |  |  |
| * Work from right to left through a network |  |  |  |  |
| * Use the table method to find the stage, state, action, destination and value |  |  |  |  |
| * Maximise the minimum and minimise the maximum   Trace back to find and state the route |  |  |  |  |
| * Re-draw diagrams so that any vertical lines go left to right |  |  |  |  |
| * Draw the decision network where edges represent actions |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Red** | **Amber** | **Green** | **To address this before the exam I will:-** |
| **Critical Path Analysis** |  |  |  |  |
| * Able to model a project using an activity network from a precedence table |  |  |  |  |
| * Able to identify where dummies are needed because of precedence |  |  |  |  |
| * Able to identify where dummies are needed to show unique task |  |  |  |  |
| * Able to carry out a forward pass to find the early event times |  |  |  |  |
| * Able to carry out a backward pass to find the latest event times |  |  |  |  |
| * Able to identify the critical events |  |  |  |  |
| * Able to identify the critical activities and the critical path |  |  |  |  |
| * Able to construct a cascade (Gantt) chart |  |  |  |  |
| * Able to use a cascade (Gantt) chart |  |  |  |  |
| * Able to calculate the minimum number of workers needed from the critical path time |  |  |  |  |
| * Able to calculate the minimum number of workers needed from the cascade (Gantt) chart |  |  |  |  |
| * Use and construct a resource diagram |  |  |  |  |
| * Use and construct a levelling diagram |  |  |  |  |
| **Network Flows** |  |  |  |  |
| Understand and use source and sinks |  |  |  |  |
| * Understand flow is less than or equal to edge capacity |  |  |  |  |
| * Understand the principle of the vertex condition, inflow = outflow |  |  |  |  |
| * Be able to augment flow paths |  |  |  |  |
| * Understand the concept of saturated edges |  |  |  |  |
| * Evaluate a cut; add the capacities |  |  |  |  |
| * Use the Max flow/Min cut theorem |  |  |  |  |
| * Understand and use the correct labelling procedure: excess flow or forward capacity, back flow or reverse capacity |  |  |  |  |
| * Super-sources and super-sinks |  |  |  |  |
| * Resolving bottlenecks |  |  |  |  |
| **Linear Programming: The Simplex Algorithm** |  |  |  |  |
| * Be able to formulate a linear programming problem into a tableau |  |  |  |  |
| * Introduce slack variables |  |  |  |  |
| * Use the ratio test to select the pivot |  |  |  |  |
| * Perform the simplex algorithm to solve the problem |  |  |  |  |
| * Interpret the tableau solution |  |  |  |  |
| **Game Theory for Zero Sum Games** |  |  |  |  |
| * Understand what is meant by zero sum games |  |  |  |  |
| * Set up a pay off matrix |  |  |  |  |
| * Use Row minimum and column maximum to find saddle points |  |  |  |  |
| * Find the value of a game |  |  |  |  |
| * Understand play safe strategies |  |  |  |  |
| * Understand stable solutions and saddle points |  |  |  |  |
| * Reduce a matrix due to row or column dominance |  |  |  |  |
| * Find solutions to mixed strategy games |  |  |  |  |
| * Find the feasible region |  |  |  |  |

|  |
| --- |
| **REVISION**  **Use the information on this checklist to make revision cards and notes** |

**Grade tracking:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Grade* | *Date* | *Grade* | *Date* | *Grade* | *Date* |
|  |  |  |  |  |  |
| *Grade* | *Date* | *Grade* | *Date* | *Grade* | *Date* |
|  |  |  |  |  |  |

*Note: You should discuss this checklist regularly with your subject teacher/mentor*