Group Contribution Statement:

Group Contribution Statement

Group Members: (Id)

- **Boran - 750043835**
- **Henry - 750028696**
- **Howard -750025583**
- **Theo - 750041957**
- **Steve - 750039082**

Boran

- Designing and implementing the brute-force solution in 'brute_force.py'
- Writing the explanation and pseudocode for the brute-force approach.
- Ensuring correctness of exhaustive subset evaluation
- Collaborating with Theo to refine and test the brute-force implementation
- Writing the explanation and pseudocode for the dynamic algorithm.

Henry

- Designing and implementing the overall file structure of the project
- Creating and organising input files for testing
- Implementing input file reading and parsing into Python
- Writing class for an activity
- Designing and implementing the dynamic programming solution and dynamic with both constraints
- Final organisation and submission of project
- Contributed to the README.md in Code/README.md

Howard

- Comparing algorithm performance across varying input sizes
- Wrote and refined the majority of written documentation by ensuring clarity, consistency, and tone
- Integrated and restructured team members' notes into the final report
- Developed and refined complexity analysis and theoretical comparisons between brute-force and dynamic programming approaches.
- Leading presentation preparation and structured slide content
- Coordinated planning and organisation for the final documentation and PowerPoint

Theo:

- Designed and created brute force and enhanced brute force algorithms, Boran assisted with designing the algorithm itself and I implemented it into the code
- Tested brute force and enhanced brute force algorithms
- Created output structure in main program
- Created system to time measure algorithm execution time
- Improved pseudocode readability
- Assisted with documentation refinement
- Contributed to the README.md in Code/README.md

Steve

- Implemented the multiple constraints (time and budget)
- Modified the planner to ensure selected activities satisfy both constraints simultaneously.
- Updated the enhanced brute-force implementation accordingly
- Assisted in adapting the dynamic programming approach to handle dual constraints
- Created and tested input cases where both constraints are binding
- Contributed to documentation explaining how adding multiple constraints affects algorithm complexity