# Summery

This paper is about using big AI language models (LLMs like GPT-3/4) together with a planning method called Monte Carlo Tree Search (MCTS). Instead of just picking the next best step, MCTS explores lots of possible action sequences, and the LLM helps suggest smart next steps at every point. The combination does a better job with multi-step tasks that need common sense, like planning or solving puzzles, compared to just using an LLM by itself.

## **What I Did in My Code:**

- I made a simple version of their method for making a daily to-do plan.
- The “LLM” part is a Python function that suggests reasonable next steps, just like a real LLM would, but faster and free.
- MCTS tries out different sequences of these suggestions, simulates rewards, and finds the best plan for that day.
- My reward score goes up for plans that include goals and have more variety, just like the paper rewards better solutions.

## **Results:**

- My code finds good, multi-step daily plans, like ['Exercise', 'Attend lecture', 'Wake up', 'Eat breakfast', 'Check email'], instead of just one action.
- The plot shows how the program gets better at planning over time as it explores more options.
- It works the way the paper describes: searching for the best sequence, not just the best single step.

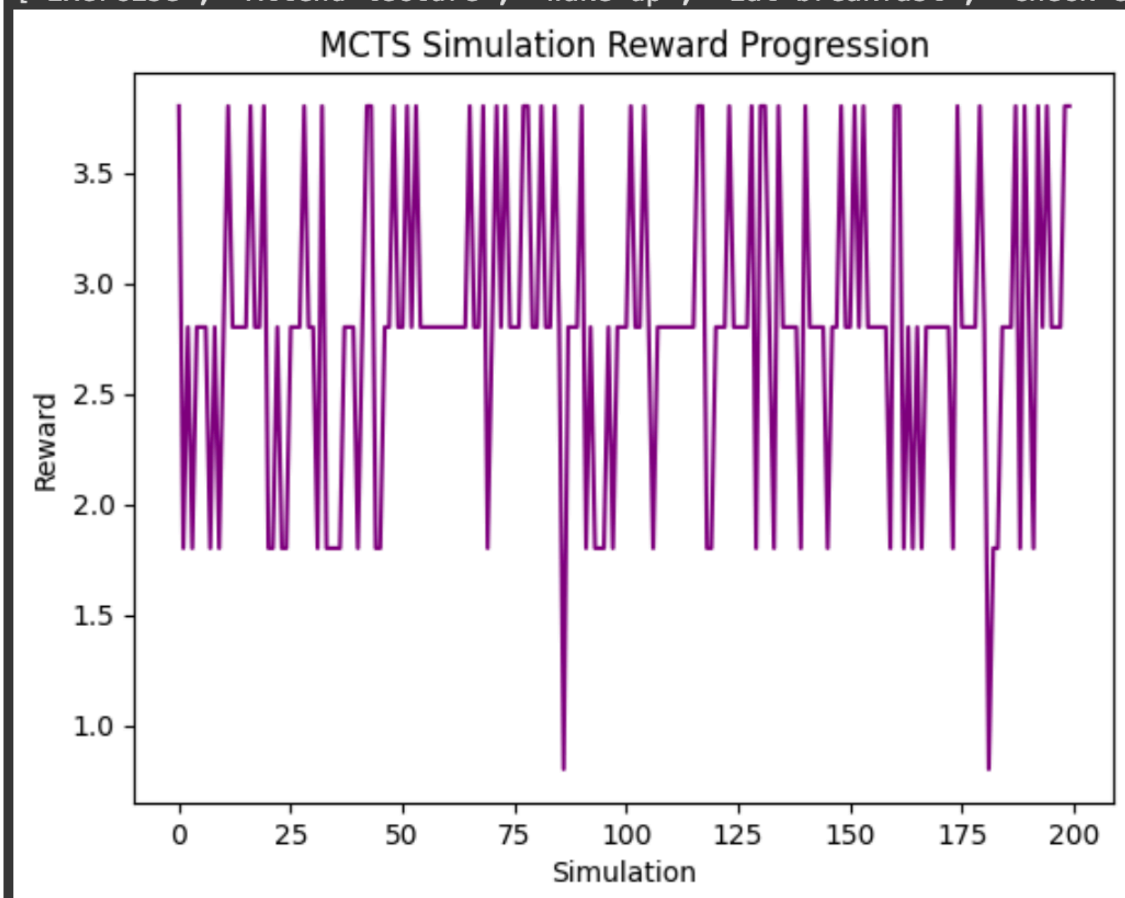
## **Conclusion:**

This matches the paper’s idea: MCTS + LLM together work better for planning than either alone. My example is simple, but shows how exploring several sequences (not just the “next best move”) leads to smarter plans, just like the authors found in their research.

## Screenshots outputs for Evidence:

```
Option 1: Plan = ['Wake up'], Value = 2.70
Option 2: Plan = ['Eat breakfast'], Value = 2.77
Option 3: Plan = ['Exercise'], Value = 2.94
Option 4: Plan = ['Check email'], Value = 2.70
Option 5: Plan = ['Work on assignment'], Value = 2.74

Full best plan found by MCTS:
['Exercise', 'Attend lecture', 'Wake up', 'Eat breakfast', 'Check email']
```




**Hasnain Khaja**
**00244783**