Part 1:

Updated Excel Spreadsheet:

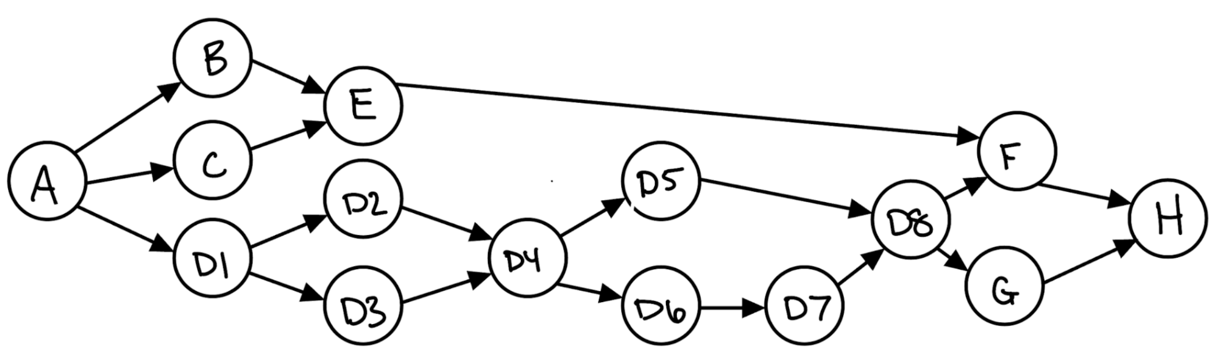
taskID	task	predecessorTask	bestCaseHours	expectedHours	worstCaseHours	projectManager	frontendDevelop	backendDevelop	dataScientist	dataEngineer
A	Describe product		2	4	8	1	0	0	0	0
B	Develop marketing strategy		8	12	16	1	0	0	0	0
C	Design brochure	A	3	5	8	1	0	0	0	0
D	Develop product prototype									
D1	Requirements ar	A	10	15	20	1	1	0	1	1
D2	Software design	D1	15	20	25	0	1	2	0	1
D3	System design	D1	16	24	32	1	1	2	0	0
D4	Coding	D2, D3	40	60	80	1	2	2	1	1
D5	Write document	D4	8	12	16	1	1	1	1	1
D6	Unit testing	D4	16	24	32	1	1	1	1	1
D7	System testing	D6	24	36	48	1	1	1	1	1
D8	Package delivera	D5, D7	4	8	12	1	1	1	1	1
E	Survey potential	B, C	12	18	24	1	0	0	1	0
F	Develop pricing	D8, E	8	12	16	1	1	1	1	1
G	Develop implem	A, D8	16	24	32	1	1	1	1	1
H	Write client prop	F, G	8	12	16	1	0	0	0	0

Many of these are guesses or estimations, but this should be a good basis point for the scheduling program. It is assumed that the Project Manager is included in almost every task to help every member stay on track and within the project parameters.

Uncertainties:

Working from my experience, the largest uncertainty with any project comes from existing requirements being altered or entirely new ones being introduced after work on the project has begun. While this is unfortunate, the best way to avoid it is to work with the client as closely as possible during the planning phase of the project. Another unknown is how the team will work together. Specifically, because this team has never worked together before, the team's “chemistry” is unknown, so people are not aware of each others’ strengths and weaknesses. An additional unknown comes from the time estimates. It is incredibly difficult to accurately predict how long a task may take, especially when the abilities of the other contributors are not known. At work if we have a hefty task, when estimating the time, I may say it will take an hour for me to complete if my first idea works, otherwise it’s anybody’s guess.

Directed Graph:



The directed graph above matches the excel sheet, with the exception that B requires A to be complete, as I believe it is not possible to create a marketing strategy if we have not yet described the product being marketed. Additionally, the graph above highlights which tasks can be completed in parallel. B, C, and all of D can be completed in parallel as the marketing strategy and brochure have little to do with the actual implementation of the program. Likewise, E can also be completed with D (but after B and C). Within the prototyping step, D2 and D3 can be completed in parallel and D5 with D6 and D7. After the prototype is complete, the development of a pricing plan and an implementation plan can be done in parallel.

Part 2:

Linear Programming Model:

Goal: Minimize the total time necessary for the project

Let T be the set of all tasks

`T = {A, B, C, D1, D2, D3, D4, D5, D6, D7, D8, E, F, G, H}`

Objective Function: We assume that each worker charges the exact same rate, so in optimizing time we will also be optimizing cost.

Minimize the total time i.e. `start_time_H - start_time_A`

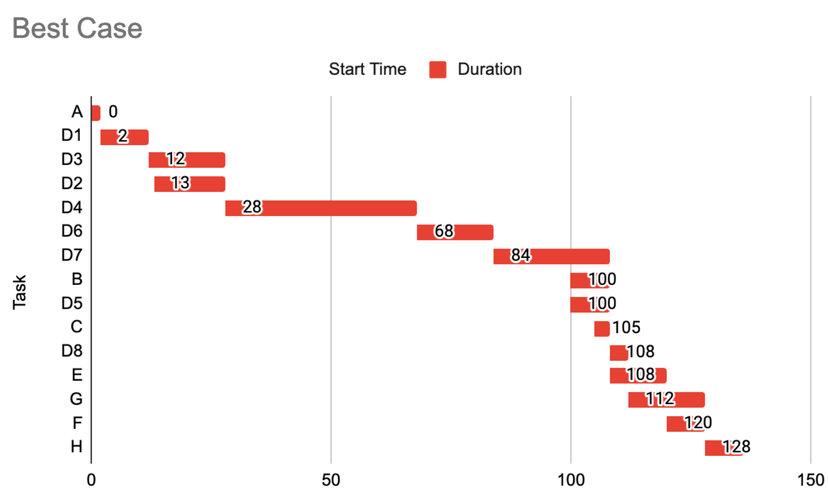
Constraints: No task may begin until its necessary predecessors are complete. Using the directed graph above, the following constraints are created:

```
start_time_B >= start_time_A + duration_A
start_time_C >= start_time_A + duration_A
start_time_D1 >= start_time_A + duration_A
start_time_G >= start_time_A + duration_A
start_time_E >= start_time_B + duration_B
start_time_E >= start_time_C + duration_C
start_time_D2 >= start_time_D1 + duration_D1
start_time_D3 >= start_time_D1 + duration_D1
start_time_D4 >= start_time_D2 + duration_D2
start_time_D4 >= start_time_D3 + duration_D3
start_time_D5 >= start_time_D4 + duration_D4
start_time_D6 >= start_time_D4 + duration_D4
start_time_D7 >= start_time_D6 + duration_D6
start_time_D8 >= start_time_D5 + duration_D5
start_time_D8 >= start_time_D7 + duration_D7
start_time_F >= start_time_D8 + duration_D8
start_time_F >= start_time_E + duration_E
start_time_G >= start_time_D8 + duration_D8
start_time_H >= start_time_F + duration_F
```

```
start_time_H >= start_time_G + duration_G
```

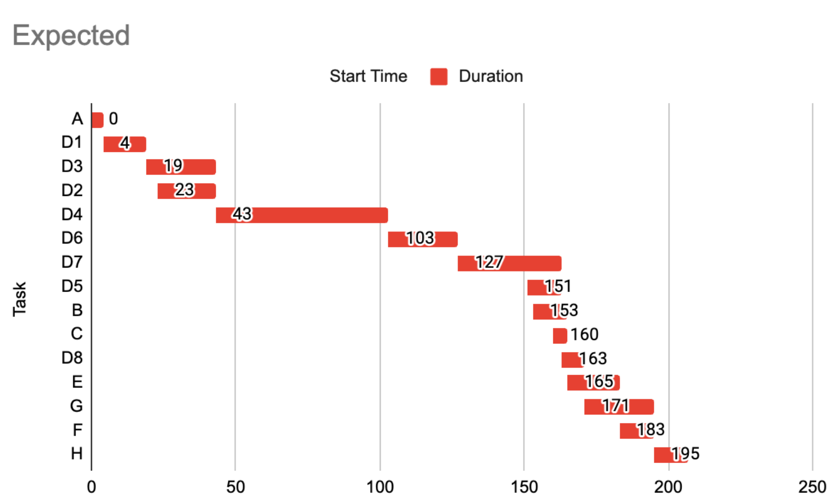
Part 3:
See Github repository: https://github.com/jbaker12/MSDS_460_Transshipment

Part 4:
Best Case Scenario:
Gantt Chart:



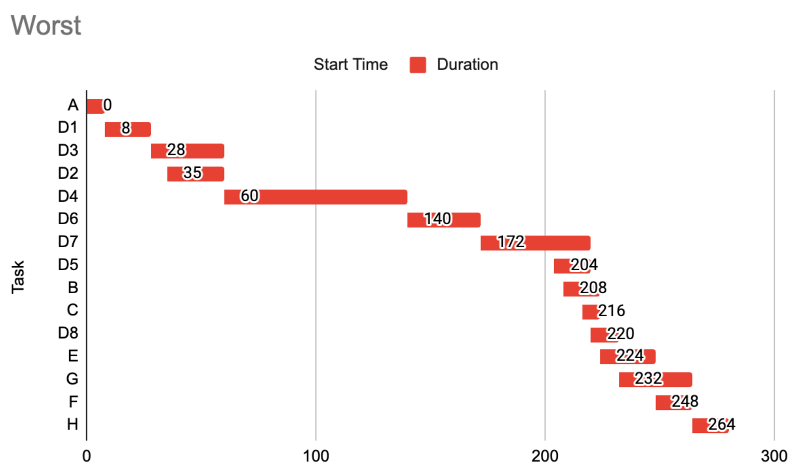
Critical Path:
A→D1→D3→D4→D6→D7→D8→G→H

Expected Scenario:
Gantt Chart:



Critical Path:
A→D1→D3→D4→D6→D7→D8→G→H

<u>Worst Case Scenario:</u>
Gantt Chart:



Critical Path:
A→D1→D3→D4→D6→D7→D8→G→H

For all three scenarios, what clearly holds up progress on the overall project is the development of the prototype, specifically, the coding of it. All of the subtasks of D were included in each critical path analysis with the exception of Software Design and Documentation, as they are the least time consuming task that is completed in parallel with another. Additionally, it is important to note that this sort of sequential process is not entirely realistic. For example, a good developer will work on documentation while developing the code, as well as testing. In fact, a practice becoming more commonplace is test driven development where a comprehensive test suite is written *first* then the functional code is written to satisfy the tests.

Part 5:
For the purpose of this proposal, we will use a standard charge of $60/hour for each staff member. It is also important to outline the specific meaning of the hour estimations with the number of workers. We will treat the number of hours represented in the cell as the total number of man hours needed for the project. Moreover, if a task is listed as requiring 20 hours with 1 frontend developer, one backend developer, one project manager, and one data engineer, then this means all of their hours add up to 20, so we will charge them $1,200 for that task. To have a robust payment schedule that accounts for the worst case scenario, we will charge the client for the worst case timing. With the duration outlined above, this comes out to a total of 280 hours, or a charge of $16,800. We will give the company a 2 week worst-case timeline for our development. Additional engineers could help speed up the timeframe, but at some point there are too many cooks in the kitchen and they would not be of any additional assistance.

As any programmer knows, sometimes a project takes significantly longer than expected. As such, it could be useful to use a Monte Carlo simulation to vary the worst case scenarios to simulate extensive hang ups on certain tasks. Also, instead of simply guessing the worst case scenario for the purpose of the cost estimation, we could utilize stochastic programming. In doing so, we can create a normal distribution centered around the expected hours for each task to better account for the randomness of real world implementation.