Hacettepe University - Computer Engineering

BBM204 Software Practicum II - Spring 2022

PROGRAMMING ASSIGNMENT 3

A JOURNEY BEYOND THE STARS

HACETTEPE COMPUTER ENGINEERING - BBM 204 - SPRING 2022

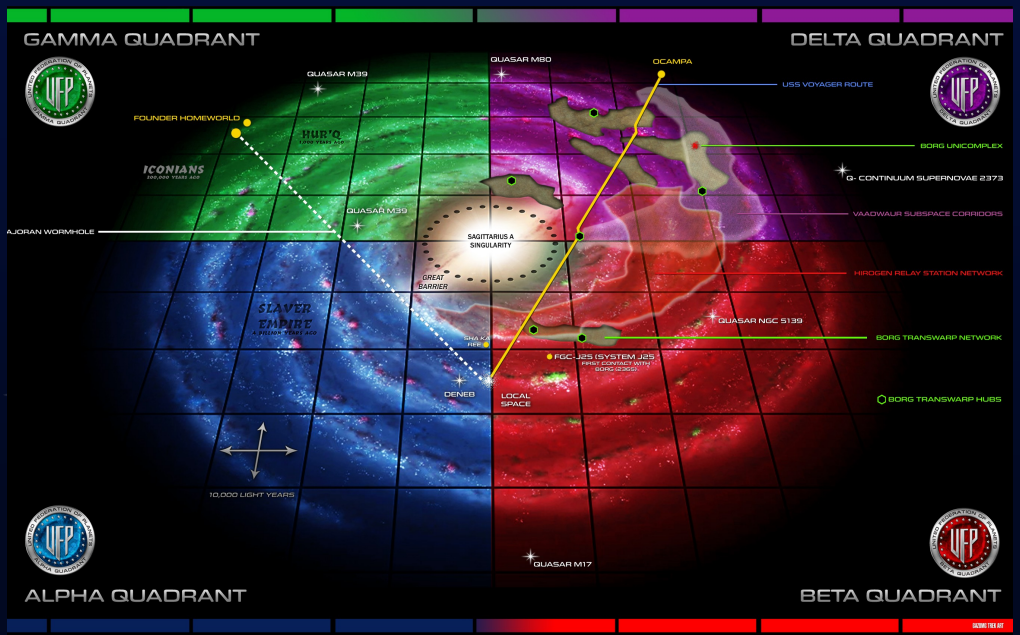
PROGRAMMING ASSIGNMENT 3 - DEADLINE 20/04/2022 AT 23:59

Topics: Graphs - Connected Components, Topological Sort, Minimum Spanning Trees

Course Instructors: Assoc. Prof. Dr. Erkut Erdem, Prof. Dr. Suat Özdemir, Asst. Prof. Dr. Adnan Özsoy
TAs: Alperen Çakın, Selma Dilek
Programming Language: Java 1.8.0
Due Date: **Wednesday, 20.04.2022 (23:59:59)**

An Expedition to Faraway Stars

The United Federation of Stars (UFS) is an interstellar union of a number of advanced planetary nation-states, founded on the principles of liberty, equality, peace, justice, and progress, with the purpose of furthering the universal rights of all sentient life. The Federation members exchange knowledge and resources to facilitate peaceful cooperation, scientific development, space exploration, and mutual defense. The Federation is known to rely on the most skilled and clever engineers in the universe to develop new technologies necessary for making new expeditions to faraway solar systems to be able to encompass the universe with justice and peace. As a skillful engineer from Earth, you are selected as one of the best programmers in the galaxy to serve for the greater good of all Federation members.



The *Milky Way* galaxy is partitioned into 4 quadrants. On an unrelated mission, a ship with the name of *UFS Voyager* was stranded in the *Delta Quadrant* due to getting badly damaged during its transportation. This tragic incident caused the ship to spend 7 years in the *Delta Quadrant* and they had a lot of stories to tell when they finally returned. Upon hearing about the troubles faced by *UFS Voyager* during its time in the *Delta Quadrant*, the Federation swore an oath to explore this quadrant, down to every piece of floating rock.

Your mission is to help the Federation by designing software that will be used both in the planning and the aftermath of this important expedition into this dangerous and unknown quadrant.

1 Part I - Mission Groundwork

The first mission of this expedition is to get some groundwork done before humans can successfully embark on the deep-space travel. This mission will involve development and building of some crucial tools, equipment, devices, and vehicles necessary for the missions in space. Through careful work of cost analysts employed by the Federation, you have been given a list of projects and their related tasks. You are expected to schedule those tasks for each project, and create a timetable for the construction and engineering teams.

1.1 Background Information and Objectives

One of those projects is building a space probe. The project task information will be given as follows: for each task you will be given the task ID, description, duration in days, and the list of tasks that must be completed before that task can commence (see Table below).

Project: Space Probe Construction			
Task ID	**Task Description**	**Duration**	**Depends on**
0	Space Probe Hull Construction	5 days	-
1	Hardware and Operating System Installation	4 days	0
2	Radar Installation	2 days	1
3	Navigation Module Installation	1 days	2, 4
4	Antenna and Communication HW/SW Installation	4 days	0, 1
5	Testing and Finalizing Project	3 days	3, 4

To understand what each row means, you can consider the row with Task 3: Task 3 involves installation of a navigation module, it will take 1 day to complete, and it cannot start unless both Tasks 2 and 4 are finished. In other words, Task 3 depends on the completion of Tasks 2 and 4. You can assume that in a project with N tasks, Task 0 will never have any predecessors (it will not depend on any other task), and that Task $N-1$ will not be a predecessor to any other task (no other task will depend on Task $N-1$). For all remaining tasks, you cannot make any similar assumptions. They will be given in a random order (e.g., you can observe that Task 3 depends on Task 4 in Project Space Probe Construction). Only Task 0 can be assumed to be the *START* task, and Task $N-1$ to be the *END* task.

Each task will be handled by a different construction or engineering team, and since some tasks are dependent on the completion of others, the assigned teams need to know when their work can begin in order to ensure efficient work planning, while at the same time the project managers need to know the completion time for each project for efficient team assignment and overall mission planning. **The goal of this mission is to schedule all projects and tasks within them such that they will be completed as early as possible.**

> **Mission Objective**:
> *Obtain a task schedule for each given project such that each task is scheduled at the earliest time possible given the dependency constraints among the tasks within a project.*

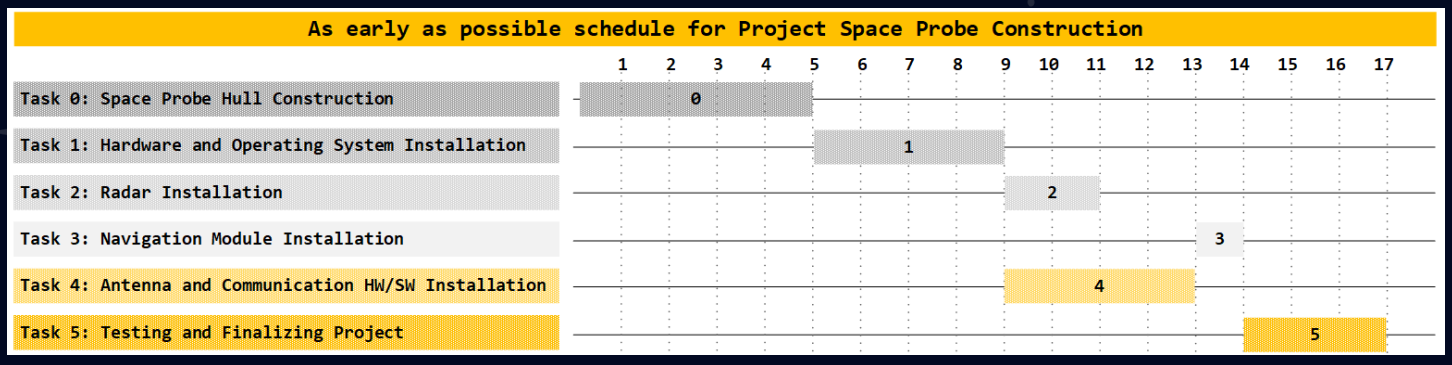
1.2 Input File Format

The input file with all projects will be given in the XML format as the *first command line argument*. Your program should parse all the projects into a *List* of *Projects* for further operations. The input file format is shown below:

```
<Projects>
    <Project>
        <Name>Space Probe Construction</Name>
        <Tasks>
            <Task>
                <TaskID>0</TaskID>
                <Description>Space Probe Hull Construction</Description>
                <Duration>5</Duration>
                <Dependencies></Dependencies>
            </Task>
            <Task>
                <TaskID>1</TaskID>
                <Description>Hardware and Operating System Installation</Description>
                <Duration>4</Duration>
                <Dependencies>
                    <DependsOnTaskID>0</DependsOnTaskID>
                </Dependencies>
            </Task>
            ...
        </Tasks>
    </Project>
    <Project>
        <Name>Space Shuttle Construction</Name>
        <Tasks>
            <Task>
                <TaskID>0</TaskID>
                <Description>Space Shuttle Hull Construction</Description>
                <Duration>10</Duration>
                <Dependencies></Dependencies>
            </Task>
            ...
        </Tasks>
    </Project>
</Projects>
```

1.3 Expected Solution and Output Format

For the given example project *Space Probe Construction*, the expected solution schedule in which every task is scheduled at the earliest day possible, and which ensures the earliest completion of the project, is given in the figure below.



From the project timeline you can observe that Tasks 0, 1, 3, 4, and 5 are on so called **critical path** of the project schedule, because delaying any of those tasks (moving their start days) would result in the overall increase of the project completion time. On the other hand, Task 2 can be scheduled to start on either of Days 9, 10, or 11 and the overall project completion time still would not change. Such tasks are called *mobile* tasks. The mobility of the tasks on the critical path is always 0. You are expected to obtain the project schedule such that all tasks are scheduled at the earliest possible times (even the mobile tasks). The earliest task start times, their end times, and the total duration of the example project are given below:

Task ID	0	1	2	3	4	5
Start time	0	5	9	13	9	14
End time	5	9	11	14	13	17

Minimum project duration: 17 days

You are expected to complete:

- ***readXML()*** *and* ***printSchedule()*** *functions from the class* ***MissionGroundwork****,*
- ***getEarliestSchedule()*** *and* ***getProjectDuration()*** *functions from the class* ***Project****.*

The expected STDOUT output format for a single project schedule is given below. Each project schedule should be printed in the same format one after another without extra blank lines. Note that your output must match the given output format for full credit.

```
### MISSION GROUNDWORK START ###
-------------------------------------------------------------------
Project name: Space Probe Construction
-------------------------------------------------------------------
Task ID   Description                                  Start  End
-------------------------------------------------------------------
0         Space Probe Hull Construction                0      5
1         Hardware and Operating System Installation   5      9
2         Radar Installation                           9      11
3         Navigation Module Installation               13     14
4         Antenna and Communication HW/SW Installation 9      13
5         Testing and Finalizing Project               14     17
-------------------------------------------------------------------
Project will be completed in 17 days.
-------------------------------------------------------------------
### MISSION GROUNDWORK END ###
```

MISSION GROUNDWORK COMPLETE

2 Part II - Mission Exploration

After a hard and professional work of all engineers involved in Mission Groundwork, all necessary space vehicles, equipment, and machinery were completed and ready to be employed in the deep-space expedition. Exploration Mission into the Delta Quadrant was officially launched with the goal of discovering and mapping new celestial bodies and planetary systems in the Delta Quadrant, which were unknown prior to the mission.

2.1 Background Information and Objectives

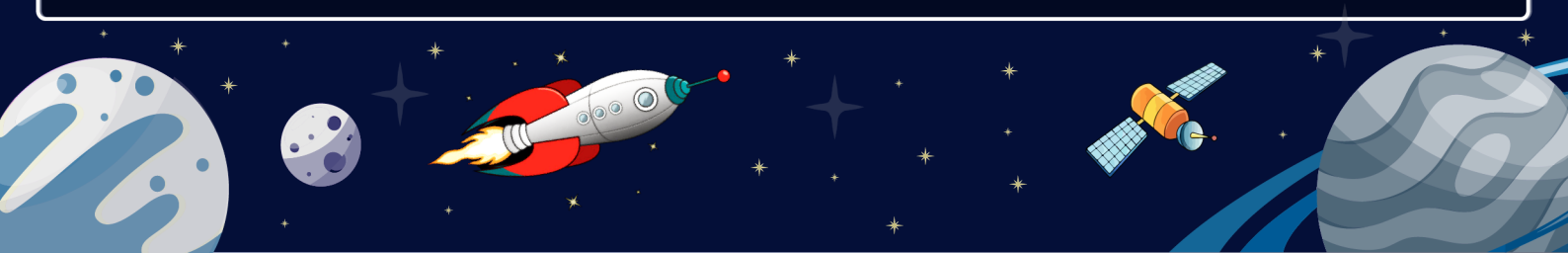
The space probes that were deployed to collect data were not equipped with advanced software which could process data. For the sake of preserving energy, they were only equipped with minimum scanning, storage, and computation capabilities, enough to collect and store information about each encountered planet. The collected data includes a planet's technological advancement level, and a list of its closest neighboring planets (other planets within its communication range). The discovered planets were assigned IDs as their official names are not known yet. **As you can see, we do not have much information about the planets, but we do know that if any two planets are neighbors, they must be inside the same solar system.**

> **Mission Objective**:
> *Given the records of the discovered planets obtained during the exploration mission in the Delta Quadrant, design software to identify the number of different solar (planetary) systems and the planets in each newly discovered solar system.*

2.2 Input File Format

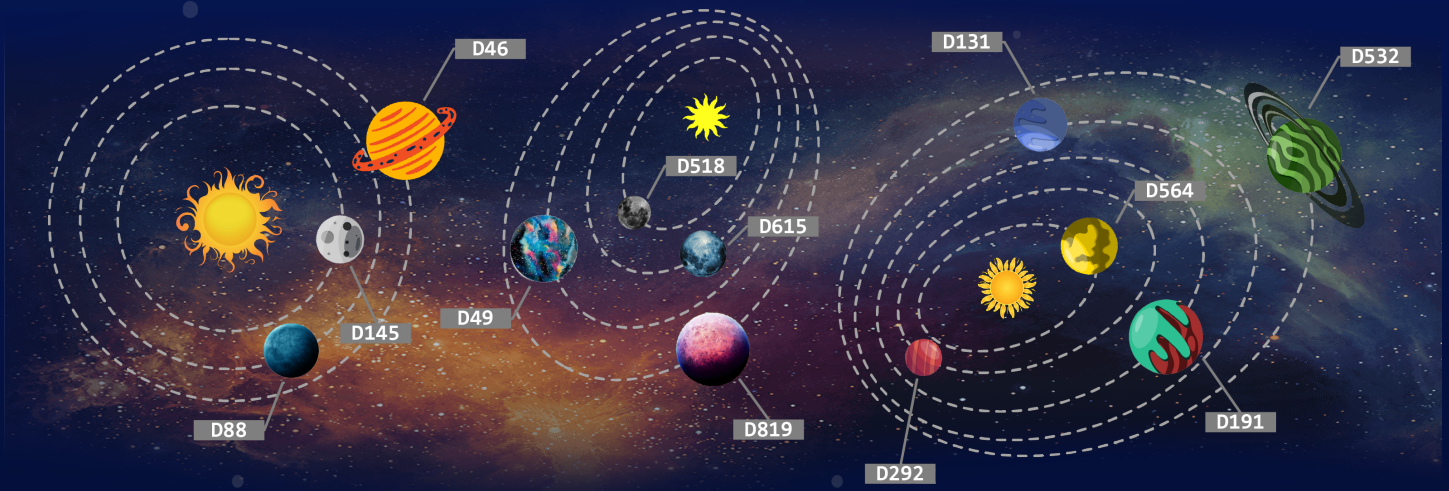
The input file with data for all discovered planets will be given in the XML format as the *second command line argument*. Your program should parse all the planets into an instance of *Galaxy* class for further processing. The input file format is given below:

```
<Galaxy>
    <Planet>
        <ID>D145</ID>
        <TechnologyLevel>6</TechnologyLevel>
        <Neighbors>
            <PlanetID>D46</PlanetID>
            <PlanetID>D88</PlanetID>
        </Neighbors>
    </Planet>
    <Planet>
        <ID>D46</ID>
        <TechnologyLevel>3</TechnologyLevel>
        <Neighbors>
            <PlanetID>D145</PlanetID>
        </Neighbors>
    </Planet>
    <Planet>
        <ID>D88</ID>
        <TechnologyLevel>2</TechnologyLevel>
        <Neighbors>
            <PlanetID>D145</PlanetID>
        </Neighbors>
    </Planet>
    ...
</Galaxy>
```



2.3 Expected Solution and Output Format

For the given sample input file, the expected solution is illustrated below. We observe that three new solar systems were found with some planets orbiting around their respective stars.



You are expected to complete:

- ***printSolarSystems()*** *and* ***readXML()*** *functions from the class* ***MissionExploration****,*
- ***exploreSolarSystems()*** *function from the class* ***Galaxy****.*

The expected STDOUT output format for the given sample input is given below. Note that your output must match the given output format for full credit; however, your numbering of the discovered solar systems may differ, but the planet list must be printed sorted according to planet IDs in ascending order.

```
### MISSION EXPLORATION START ###
3 solar systems have been discovered.
Planets in Solar System 1: [D46, D88, D145]
Planets in Solar System 2: [D49, D518, D615, D819]
Planets in Solar System 3: [D131, D191, D292, D532, D564]
### MISSION EXPLORATION END ###
```

MISSION EXPLORATION COMPLETE

3 Part III - Mission Networking

In the final mission, you are assigned a task to design a cost-efficient subspace communication network between the newly discovered solar systems, so that no other Federation starship will ever get stranded in the *Delta Quadrant* without being able to send at least a distress message. Subspace communication enables sending data and messages across interstellar distances faster than the speed of light, by transmission of a subspace radio signal, which travels through subspace rather than normal space. These subspace channels are called hyperchannels.

3.1 Background Information and Objectives

The planets within the same solar system can already communicate with each other using an interplanetary communication network. To establish a hyperchannel between two solar systems, a delegate planet in each solar system must be chosen. The cost of subspace communication is inversely proportional to the collective technology levels of two communicating planets. This is because building more efficient subspace communication relays that establish hyperchannels between two solar systems is easier on a planet with a higher technological advancement level. For this reason, the planet with the highest technology level should be chosen as the delegate planet for each solar system.

The cost of subspace communication between two solar systems i and j is calculated as follows:

$$Avg_{i,j} = \frac{MaxTechLevel_i + MaxTechLevel_j}{2}$$

$$Cost_{i,j} = \frac{Subspace\ Communication\ Constant}{Avg_{i,j}}$$

where Subspace Communication Constant is 144.5.

Your mission is to design a software that will create a subspace communication network among newly discovered planetary systems. In other words, your software will determine which hyperchannels will be chosen to constitute the newly established network such that the overall communication cost will be minimum. **The network will be fully established when each solar system can be reached from any other solar system either through a direct hyperchannel between their respective delegate planets, or through a path of hyperchannels that connect them indirectly via relays in other intermediary solar systems.**

Mission Objective:
Use the information about the newly discovered solar systems, obtained in the previous mission, to establish the least-cost subspace communication network among the solar systems.

3.2 Input

Your program should use the **List of SolarSystem objects** obtained in the previous mission as the input for this mission.

3.3 Expected Solution and Output Format

Following the sample I/O from the previous mission, remember that 3 solar systems were discovered. The planets with maximum technology levels will be chosen as the delegate planets of their respective solar systems:

```
The delegate planet of Solar System 1: D88  - Technology level: 62
The delegate planet of Solar System 2: D819 - Technology level: 98
The delegate planet of Solar System 3: D532 - Technology level: 78
```

Using the formula for calculating the cost of subspace communication between two planets, we obtain the cost of each potential hyperchannel that can be considered for building the communication network:

```
Hyperchannel between D88  - D819 with cost 1.806250
Hyperchannel between D88  - D532 with cost 2.064286
Hyperchannel between D532 - D819 with cost 1.642045
```

Since it is enough to ensure the reachability between each pair of the solar systems, we should disregard building hyperchannels that will incur more unnecessary cost. In this sample scenario, a hyperchannel between D88 and D532 is not necessary because they can communicate over D819 without adding extra cost to the building of the network. The expected networking solution is illustrated below.

You are expected to complete:

- ***printMinimumCostCommunicationNetwork()*** *function from the class* ***MissionNetwork****,*
- ***getMinimumCostCommunicationNetwork()*** *function from the class* ***SubspaceCommunicationNetwork****.*

The expected STDOUT output format for the given sample input is given below. Note that the planets at the each end of a hyperchannel must be printed sorted according to planet IDs in ascending order, but the order in which you print hyperchannels is not important.

```
### MISSION NETWORKING START ###
Hyperchannel between D88 - D819 with cost 1.806250
Hyperchannel between D532 - D819 with cost 1.642045
The total cost of the subspace communication network is 3.448295.
### MISSION NETWORKING END ###
```

MISSION NETWORKING COMPLETE

HACETTEPE COMPUTER ENGINEERING - BBM 204 - SPRING 2022

PROGRAMMING ASSIGNMENT 3 - DEADLINE 20/04/2022 AT 23:59

Must-Use Starter Codes

You MUST use **this starter code**. All classes should be placed directly inside your **zip** archive. Feel free to create other additional classes if necessary, but they should also be directly inside **zip**.

Grading Policy

- Submission: 1%
- Implementation of the algorithms: 90%
  - Mission Groundwork: 30%
  - Mission Exploration: 30%
  - Mission Networking: 30%
- Output test: 9%

Important Notes

- Do not miss the deadline: **Wednesday, 20.04.2022 (23:59:59)**.
- Save all your work until the assignment is graded.
- The assignment solution you submit must be your original, individual work. Duplicate or similar assignments are both going to be considered as cheating.
- You can ask your questions via Piazza (https://piazza.com/hacettepe.edu.tr/spring2022/bbm204), and you are supposed to be aware of everything discussed on Piazza.
- You can only test your code via http://34.159.86.133/ (**does not count as submission!**).
- You must submit your work via https://submit.cs.hacettepe.edu.tr/ with the file hierarchy given below:
  - **b<studentID>.zip**
    - Constants.java <FILE>
    - Galaxy.java <FILE>
    - HyperChannel.java <FILE>
    - Main.java <FILE>
    - MissionExploration.java <FILE>
    - MissionGroundwork.java <FILE>
    - MissionNetworking.java <FILE>
    - Planet.java <FILE>
    - Project.java <FILE>
    - SolarSystem.java <FILE>
    - SubspaceCommunicationNetwork.java <FILE>
    - Task.java <FILE>
- The name of the main class that contains the main method should be **Main.java**. **You MUST use this starter code**. The main class and all other classes should be placed directly in your **zip** archive. Feel free to create other additional classes if necessary, but they should also be inside the **zip**.
- This file hierarchy must be zipped before submitted (not .rar, only .zip files are supported).
- **Usage of any external libraries is forbidden.**

Run Configuration

Your code will be compiled and run as follows:

```
javac -cp . Main.java
java -cp . Main <projectsXMLFile> <galaxyXMLFile>
```

Academic Integrity Policy

All work on assignments **must be done individually**. You are encouraged to discuss the given assignments with your classmates, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in pseudocode) **will not be tolerated**. In short, turning in someone else's work (including work available on the internet), in whole or in part, as your own will be considered as **a violation of academic integrity**. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

The submissions will be subjected to a similarity check. Any submissions that fail the similarity check will not be graded and will be reported to the ethics committee as a case of academic integrity violation, which may result in suspension of the involved students.

References

[1] Vecteezy, "Star Vectors", https://www.vecteezy.com/free-vector/star, Last Accessed: 23/03/2022.

[2] tharrison, "Vector Rocket Ship stock illustration", https://www.istockphoto.com/vector/vector-rocket-ship-gm165034094-1161747, Last Accessed: 23/03/2022.

[3] abcvector, "Communication satellite", https://www.gograph.com/clipart/communication-satellite-gg91788562.html, Last Accessed: 23/03/2022.

[4] United Federation of Planets Logo, https://seeklogo.com/vector-logo/145438/united-federation-of-planets, Last Accessed: 23/03/2022.

[5] gazomg, "Star Trek Milky Way Quadrants Map", https://www.deviantart.com/gazomg/art/Star-Trek-Milky-Way-Quadrants-Map-906016780, Last Accessed: 23/03/2022.

[6] "Nebula", https://www.pngfind.com/mpng/iimoooi_view-the-latest-nebula-space-cloud-transparent-hd/, Last Accessed: 23/03/2022.

[7] Memory Alpha, "Subspace communication", https://memory-alpha.fandom.com/wiki/Subspace_communication, Last Accessed: 25/03/2022.