Towering Ports

Group 8-10

Javier Lee 3I310 (Leader)
Koh Zhu Quan 3I317
Yu Wenqi 3I131
Soong Jia Yuan 3A125

Hwa Chong Institution (High School)

Introduction and Rationale

This project is about applying the algorithm of the famous mathematical puzzle – the "Tower of Hanoi" in a real-life industrial environment. As the second largest transshipment container port in the world, Singapore handles a huge number of containers daily. These containers may come from different origins, go to different destinations, are carried by different shipping companies at different times and have various cargo profiles which restrict where they can be placed. Within a limited piece of land as in Singapore, stacking them together to minimise space occupation and handling times remains a critical question. In this project, we will study the problem from mathematical perspectives, like using the concept of "Tower of Hanoi", and provide some insight to guide the practical port operations.

Objectives

1. To apply the concept of Tower of Hanoi to Singapore's ports.
2. To find a strategy to increase efficiency under space constraints.
3. To create a method using dynamic programming which can help improve efficiency of Singapore's ports.

Research Questions

1. What is the concept behind the Tower of Hanoi that can be applied to Singapore's ports?
2. What is a good way to place the cargo such that efficiency is ideal under space constraints?
3. How will our method help to make Singapore's port operations more efficient?

Literature Review

Tower of Hanoi

In this project, we use the Tower of Hanoi concept, however there are some assumptions made. One of the basic assumptions was that import containers would be picked up randomly. However, import loads are often distributed over many containers, which could be retrieved in any order. In this case, the actual container could be assigned to a truck in a way of minimizing the number of rehandles, when the truck enters the terminal. Then, the expected number of rehandles may become less than that estimated (K.H. Kim & H.B. Kim, 2002).

Another assumption is that container traffic would be uniformly distributed over the entire yard for import containers. However, the traffic may be concentrated on parts of the yard during a certain period of time. In this case, the length of the service area assigned to a transfer crane may be reduced, which results in the decrease in the expected travel time per move of the transfer crane. On the other hand, the higher possibility of interference between transfer cranes in the area on which the traffic is concentrated may offset the effect of the reduction in the travel time of transfer cranes.

Lastly, it was assumed that containers unloaded from different vessels would not be mixed with each other in the same bay. This policy of stacking import containers is called the segregating strategy (Castilho & Daganzo, 1993). For the same average height of stacks, the non-segregating strategy usually results in a higher number of rehandles than the segregating strategy. However, under the segregating strategy, clearing moves are needed to make room for inbound containers of other incoming vessels. Thus, in case that containers unloaded from different vessels may be mixed with each other in the same bay, the expected number of rehandles should be adjusted upward accordingly.

Ning Zhao et al. (n.d.) suggests that for container delivery sequence, the export full containers are stored at the bays with a gantry crane, while the export empty containers stored at the bays with a stacker. Due to different operation techniques, the stowage for these two kinds of situations is slightly different:

(i) Bay with a stacker

The stacker cannot operate over containers, so it must operate from the outer row near to the lane to the inner row.

(ii) Bay with a gantry crane

The gantry crane can lift vertically at a high point and operate across containers. It is very flexible when delivering containers. When making stowage, only attention should be paid to the operation techniques to reduce the container reshuffle.

The Tower of Hanoi concepts will be used, under assumptions, to find the optimal method using the gantry crane.

Dynamic Programming

Requires that a problem be defined in terms of state variables, stages within a state (the basis for decomposition), and a recursive equation which formally expresses the objective function in a manner that defines the interaction between state and stage (2021). Dynamic Programming is similar to the divide and conquer algorithm and is used to solve subproblems which make up the main problem, saving each answer to avoid repetition. Dynamic Programming is typically applied to optimization problems (Alyssa Dalton, n.d.).

The different methods of dynamic programming include precompute (Compute and store values such that they do not have to be calculated again, generally used for large number of query), recursion (Shown in Fibonacci and TOH, where a base case and a recursion sequence is required), top-down (Generally used for 1d or 2d max sum or prefix sum), and bottom-up (Generally used for 1d or 2d max sum or prefix sum).

Examples of dynamic programming: Fibonacci, Longest increasing subsequence, Longest common subsequence, Coin change problem

Methodology

1. Use dynamic programming to replicate the situation in Singapore's ports.
2. Look at the different restrictions in place when stacking cargo (eg. weight, leaving sequence).
3. Using our data to find the ultimate best solution.

Findings

Tower of Hanoi

The objective of the puzzle is to move all the discs from position A to C following two simple rules:

1. Only one disc can be moved at a time.
2. No disc can be placed on top of a disc smaller than itself.
3. There are 3 pegs, A, B and C

Algorithm to find the least number of moves required to move the discs from one tower to the other: 2^n-1, where n is the number of discs.

Application of Dynamic Programming to the Tower of Hanoi problem:

Let peg A be the start point, peg B be the intermediate point and peg C be the end point.

Define an array F[n] to represent the minimum moves for n number of discs.

F[1] = 1 since the disc can just be moved to peg C.

For n >1, we can move n -1 discs to peg B and the largest disc to peg C, then move n -1 discs to peg C.

Therefore, $F[n] = 2(F[n-1])+1$

Since F[1] can be written as $2(2^0 - 1) +1 = 2^1-1$

$F[2] = 2(F[2 - 1]) +1 = 2(2^1 - 1) +1 = 2^2 -1$

$F[n] = 2(F[n -1]) + 1 = 2^n -1$

Our Experimentation

Based on the Tower of Hanoi logic and the number of moves to solve the puzzle, we experimented with a few possible cases in this real life scenario:

Scenario 1: Both have 2 containers with the same arriving sequence.

(i) Scenario 1a: Yellow container leaves before the blue container

Port X has 2 slots.

A yellow container enters Port X followed by a blue container, and the yellow container leaves before the blue container.

There are no restrictions imposed in this scenario.

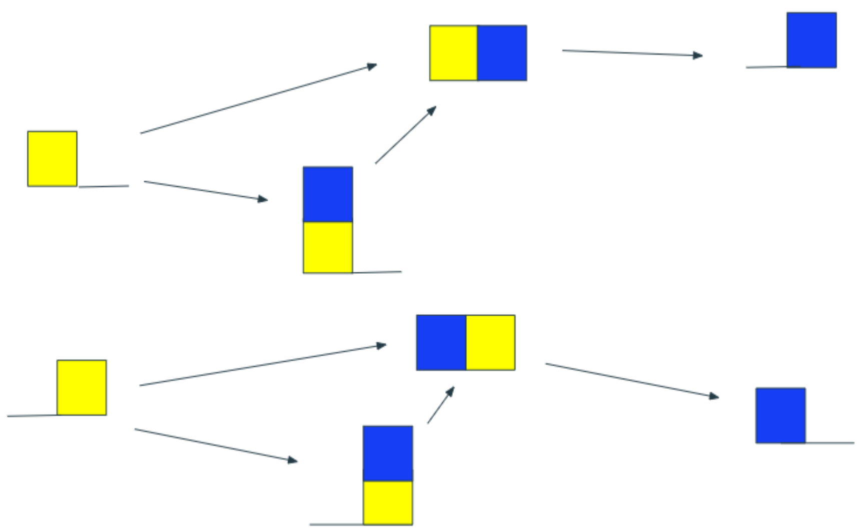

Fig. 1: Possible ways of stacking the containers for scenario 1a

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig.1. There are then 2 possible ways to stack the blue container: Either on top of the yellow container, or in the other stacking slot. However, if we were to place the blue container on top of the yellow container, the blue container would have to be shifted to the other stacking slot so that the yellow container can be unloaded. Hence, the least number of moves for this scenario is 4 moves (from empty to empty).

Conclusion for Scenario 1a:

With 2 different containers and 2 stacking slots, we would always end up with the yellow and blue containers being in two different stacking slots.

(ii) Scenario 1b: Blue container leaves before the yellow container

Port X has 2 slots.

The yellow container enters the port before the blue container and the blue container leaves before the yellow container.

There are no restrictions imposed in this scenario.

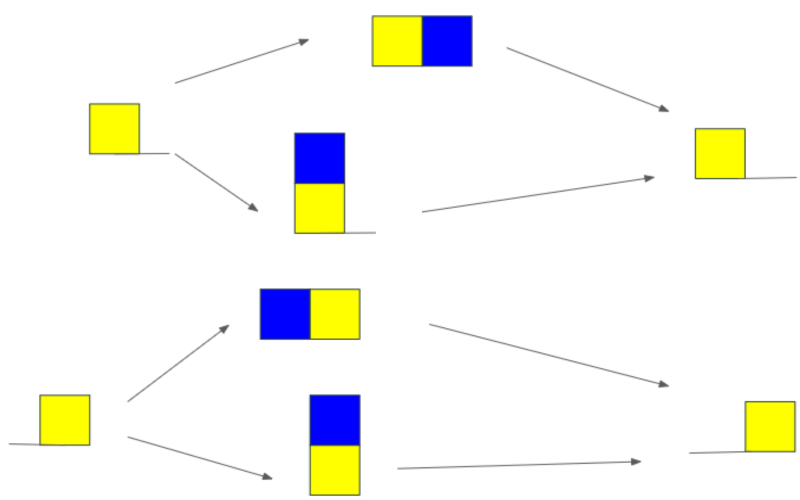

Fig. 2: Possible ways of stacking the containers for scenario 1b

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig 2. There are then 2 possible ways to stack the blue container: Either on top of the yellow container, or in the other stacking slot. No matter which way we stack the blue container, no reshuffling is needed to enable the blue container to leave the port first. Hence, the least number of moves for this scenario is 4 moves (from empty to empty).

Conclusion for Scenario 1b:

When there are only 2 containers with the container which arrives later leaving before the other, the stacking and unloading becomes very easy. No matter which slots we place the blue container in, it would always be able to leave the port without any reshuffling of containers needed.

Scenario 2: All have 3 containers with the same arriving sequence.

(i) Scenario 2a: Yellow container leaves the port first, followed by the blue container, then the red container.

Port X has 2 slots.

Yellow container enters the port first, followed by the blue container, then the red container.

There are no restrictions imposed in this scenario.

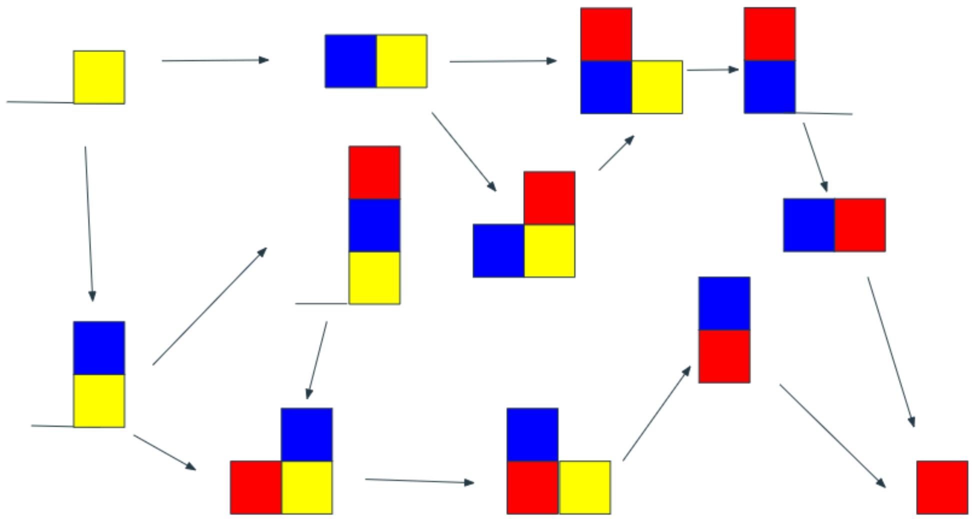

Fig. 3: Possible ways of stacking the containers for scenario 2a

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig. 3. There are then 2 possible ways to stack the blue container: Either on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can be stacked in the empty stacking slot or on top of the rest of the containers. No matter which way we choose, we would have to move the red and blue containers to the other stacking slot to enable the yellow container to be unloaded. By then, the blue container would be stacked on top of the red container and can be unloaded easily. For B, the red container can be stacked on top of the blue or yellow container. No matter how we stack the containers, we would have to move the red container on top of the blue container to allow the yellow container to be unloaded. Then, we have to move the red container to the vacant stacking slot to allow the blue container to be unloaded. Therefore, the least number of moves required for this scenario is 7 moves (empty to empty).

Conclusion for Scenario 2a:

When there are more containers than the number of stacking slots, there will definitely be containers that have to be stacked on top of one another.

(ii) Scenario 2b: The red container leaves the port first, followed by the blue container, then the yellow container.

Port X has 2 slots.

Yellow container enters the port first, followed by the blue container, then the red container.

There are no restrictions imposed in this scenario.

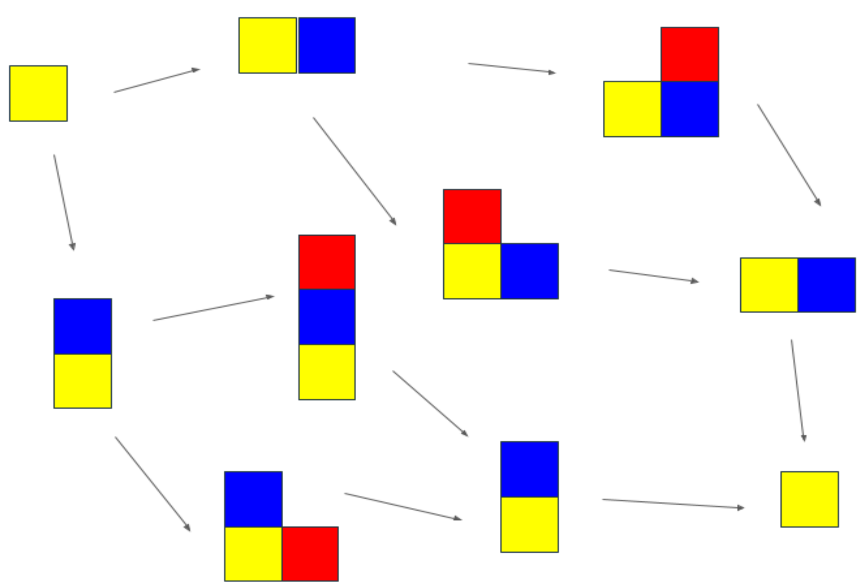

Fig. 4: Possible ways of stacking the containers for scenario 2b

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig. 4. There are then 2 possible ways to stack the blue container: Either on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can be stacked in the empty stacking slot or on top of the rest of the containers. No matter which way we choose, no reshuffling is needed to enable the red container to leave the port first, followed by the blue then yellow container. Therefore, the least number of moves required for this scenario is 6 moves (empty to empty).

Conclusion for Scenario 2b:

When there are 3 containers with the containers which arrive later leaving before the others, the stacking and unloading becomes very easy. No matter which slots we place the last container in, it would always be able to leave the port without any reshuffling of containers needed.

(iii) Scenario 2c: The yellow container leaves the port first, followed by the red container, then the blue container.

Port X has 2 stacking slots.

The yellow container enters the port first, followed by the blue container, then the red container.

There are no restrictions imposed in this scenario.

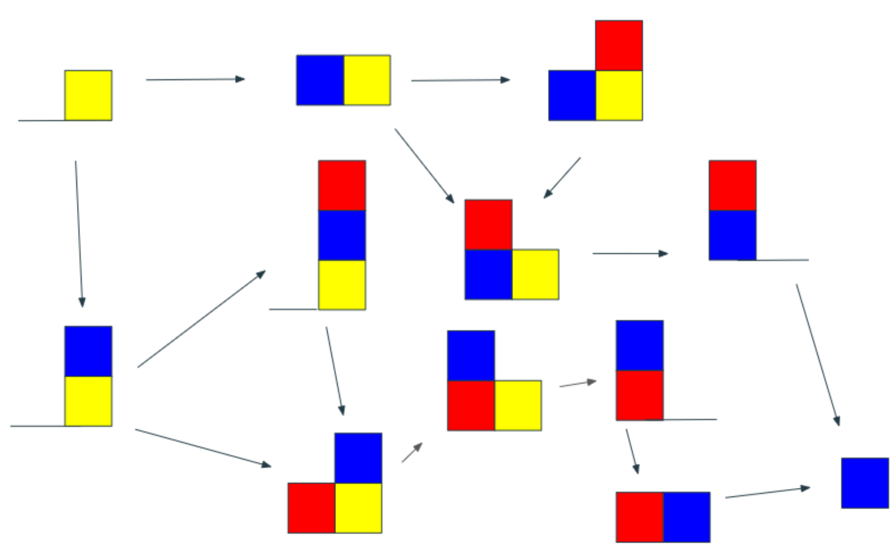

Fig. 5: Possible ways of stacking the containers for scenario 2c

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig. 5. There are 2 possible ways to stack the blue container. It can be placed on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can either be placed on top of all the containers, or in the separate stacking slot. No matter which way we choose, we would have to move the red and blue containers to the other stacking slot to enable the yellow container to be unloaded. The blue container would then have to be moved to the vacant stacking slot to enable the red container to be unloaded. The blue container can then be unloaded. For B, the red container can either be placed on top of the blue container or the yellow container. No matter which way we choose, we would have to move the red container on top of the blue container to allow the yellow container to be unloaded. Then, the blue container would have to be moved to the empty stacking slot to allow the red container to be unloaded. The blue container can then be unloaded. Hence, the least number of moves required in this scenario is 6 moves (empty to empty).

Conclusion for Scenario 2c:

When there are more containers than the number of stacking slots, there will definitely be containers that have to be stacked on top of one another.

(iv) Scenario 2d: The red container leaves the port first, followed by the yellow container, then the blue container.

Port X has 2 stacking slots.

The yellow container enters the port first, followed by the blue container, then the red container. There are no restrictions imposed in this scenario.

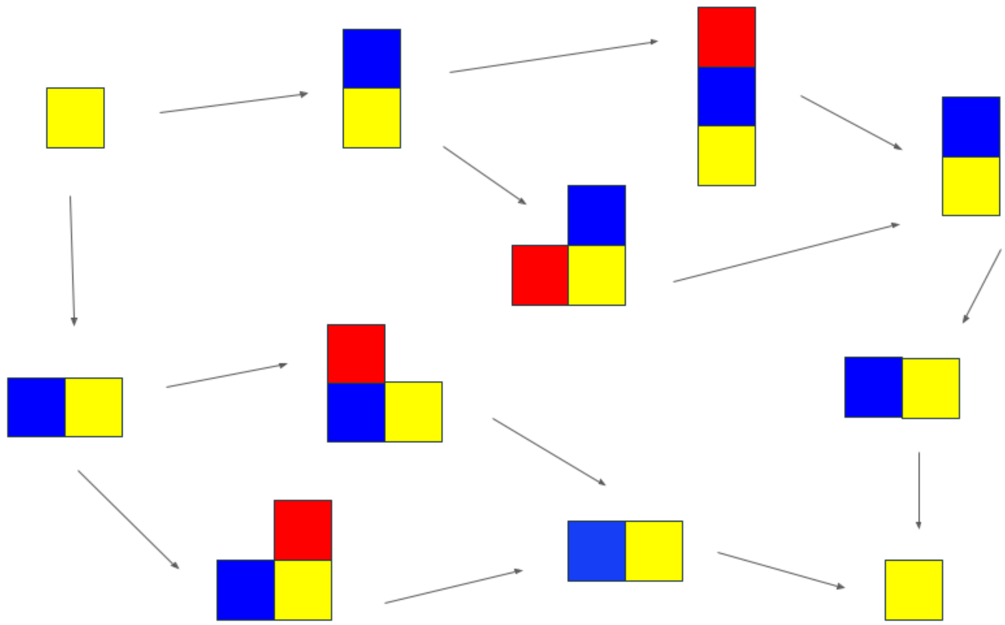

Fig. 6: Possible ways of stacking the containers for scenario 2d

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig. 6. There are 2 possible ways to stack the blue container. It can be placed on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can either be placed on top of all the containers, or in the separate stacking slot. No matter which way we choose, the red container can be unloaded without any reshuffling needed. Then, the blue container would have to be shifted to the empty stacking slot to allow the yellow container to be unloaded. The blue container can then be unloaded. For B, the red container can either be placed on top of the blue container or the yellow container. No matter which way we choose, the red container can be unloaded without any reshuffling needed. Then, the yellow container can be unloaded before the blue container easily. Hence, the least number of moves needed for this scenario is 6 moves (empty to empty).

Conclusion for Scenario 2d:

When the container which arrives last is first to leave, and we are only dealing with 3 containers with 2 stacking slots, the containers can be strategically stacked such that no reshuffling is needed to allow the containers to leave the port.

(v) Scenario 2e: The blue container leaves the port first, followed by the red container, then the yellow container.

Port X has 2 stacking slots.

The yellow container enters the port first, followed by the blue container, then the red container.

There are no restrictions imposed in this scenario.

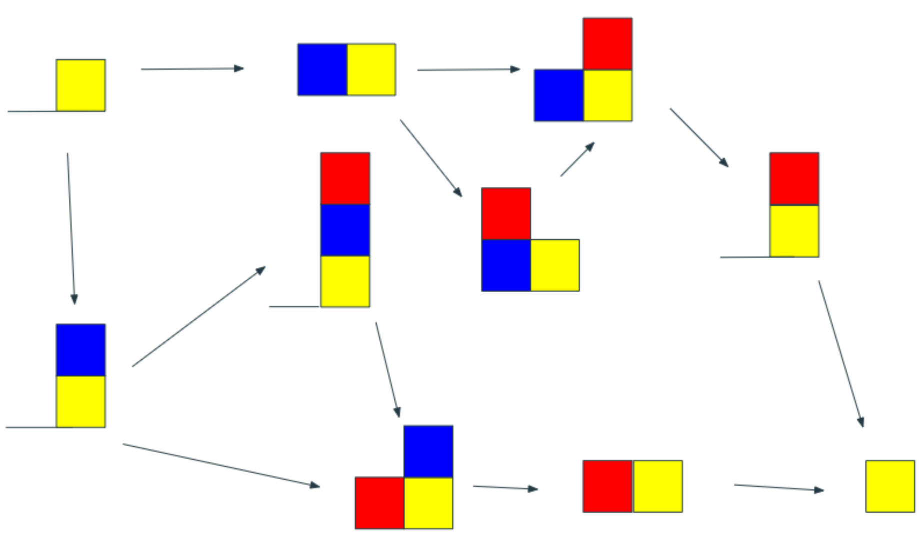

Fig. 7: Possible ways of stacking the containers for scenario 2e

Moves:

Initially, both stacking slots were empty. The yellow container arrives and is placed in one of the stacking slots, as shown in Fig. 7. There are 2 possible ways to stack the blue container. It can be placed on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can either be placed on top of all the containers, or in the separate stacking slot. Since the blue container has to leave first, then the red container can be placed in the separate stacking slot to eliminate the need to reshuffle the containers. For B, the red container can either be placed on top of the blue container or the yellow container. Since the blue container has to leave first, the red container can be placed on the yellow container as it eliminates the need to reshuffle, as the blue, then red containers can leave first before the yellow container. Hence, the least number of moves needed for this scenario is 6 moves (empty to empty).

Conclusion for scenario 2e:

When there are more containers than the number of stacking slots, there will definitely be containers that have to be stacked on top of one another. However, strategic stacking of the containers would allow us to unload the containers without any reshuffling.

(vi) Scenario 2f: The blue container leaves the port first, followed by the yellow container, then the red container.

Port X has 2 stacking slots.

The yellow container enters the port first, followed by the blue container, then the red container.

There are no restrictions imposed in this scenario.

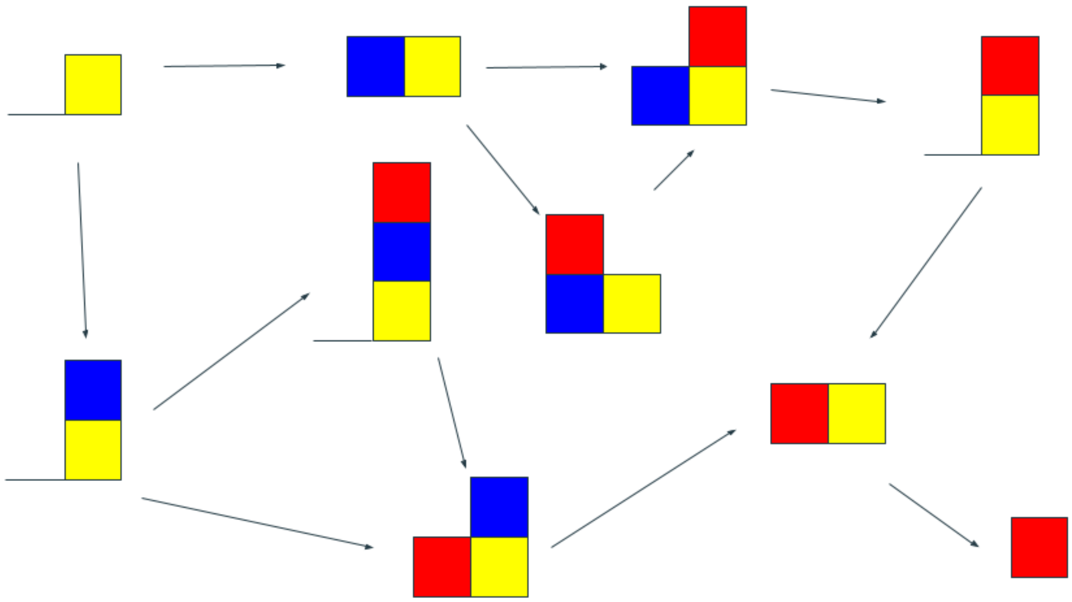

Fig. 8: Possible ways of stacking the containers for scenario 2f

Moves:

Initially, both stacking slots were empty. The yellow container enters the port and is placed in one of the stacking slots, as shown in Fig. 8. There are 2 possible ways to stack the blue container. It can be placed on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can either be placed on top of all the containers, or in the separate stacking slot. No matter which way we choose, the red container would have to be moved to a different stacking slot from the one that has the blue and yellow containers to allow the blue container to be unloaded. The red container can then be unloaded before the yellow container. For B, the red container can be stacked on top of the yellow container or the blue container. No matter which way we choose, we would have to ensure that the red container is on top of the yellow container to enable the blue container to be unloaded. The red container can then be unloaded before the yellow container. Hence, the least number of moves for this scenario is 6 moves (empty to empty).

Conclusion for scenario 2f:

When there are more containers than the number of stacking slots, there will definitely be containers that have to be stacked on top of one another. However, strategic stacking of the containers would allow us to unload the containers without any reshuffling.

Cases (Entering sequence Y-B-R)	**2a**	**2b**	**2c**	**2d**	**2e**	**2f**
Leaving Sequence	Y-B-R	Y-R-B	B-Y-R	B-R-Y	R-B-Y	R-Y-B
Steps needed (empty to empty)	7	6	6	6	6	6

Scenario 3: All have 3 containers with the same arriving sequence.

(i) Scenario 3a: The red container leaves the port first, followed by the yellow container, then the blue container.

Port X has 2 stacking slots.

The yellow container enters the port first, followed by the blue container, then the red container.

Restriction: The red container cannot be stacked on top of the yellow container

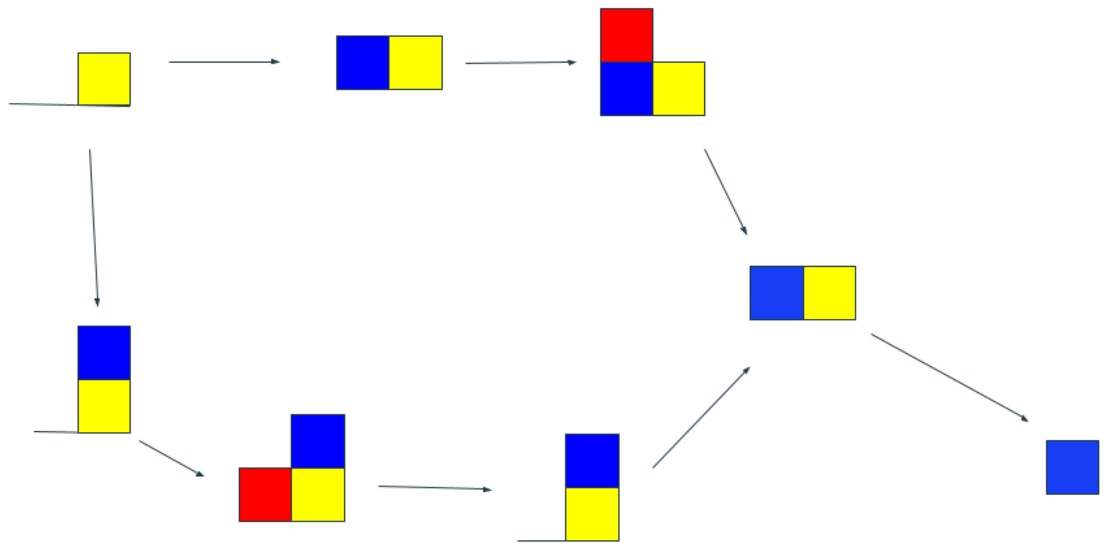

Fig. 9: Possible ways of stacking the containers for scenario 3a

Moves:

Initially, both stacking slots were empty. The yellow container enters the port and is placed in one of the stacking slots, as shown in Fig. 9. There are 2 possible ways to stack the blue container. It can be placed on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can be placed on top of the blue container or the yellow container. No matter which way we choose, the red container can be unloaded before the yellow container. For B, the red container can be stacked on top of the other stacking slot. No matter which way we choose, the red container can be unloaded before the yellow container. Hence, the least number of moves for this scenario is 6 moves (empty to empty).

Conclusion for scenario 3a:

When there are more containers than the number of stacking slots, there will definitely be containers that have to be stacked on top of one another. However, strategic stacking of the containers would allow us to unload the containers without any reshuffling. Due to some restrictions in the stacking of containers, the minimum number of moves required to load and unload the containers may differ.

(ii) Scenario 3b: The yellow container leaves the port first, followed by the blue container, then the red container.

Port X has 2 stacking slots.

The yellow container enters the port first, followed by the blue container, then the red container.

Restriction: The red container cannot be stacked on top of the yellow container

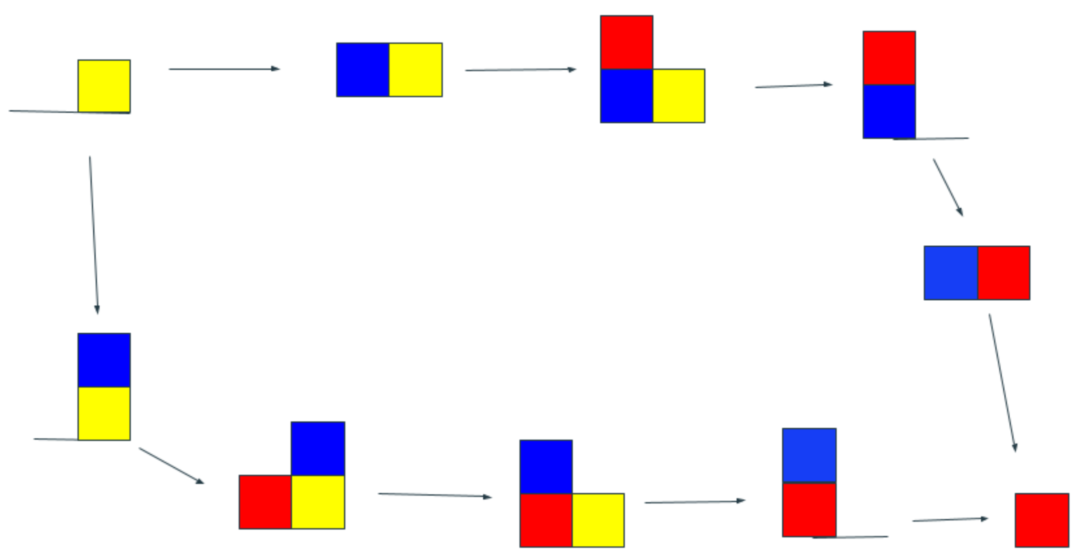

Fig. 10: Possible ways of stacking the containers for scenario 3b

Moves:

Initially, both stacking slots were empty. The yellow container enters the port and is placed in one of the stacking slots, as shown in Fig. 10. There are 2 possible ways to stack the blue container. It can be placed on top of the yellow container, or in the other stacking slot. We would label the first possible way as A and the second possible way as B. For A, the red container can be placed on top of the blue container so that the yellow container can be unloaded first. For B, the red container can be stacked on top of the other stacking slot. No matter which way we choose, there is a need to reshuffle the containers so that the red container can be unloaded first. Hence, the least number of moves for this scenario is 7 moves (empty to empty).

Conclusion for scenario 3b:

When there are more containers than the number of stacking slots, there will definitely be containers that have to be stacked on top of one another. However, strategic stacking of the containers would allow us to unload the containers without any reshuffling. Due to some restrictions in the stacking of containers, the minimum number of moves required to load and unload the containers may differ.

Conclusions:

Dynamic Programming was applied to the Tower of Hanoi to find the algorithm behind it. By using the array F[n] to be the minimum number of moves needed for n number of discs, F[n] = 2(F[n -1]) + 1. When n = 1, F[1]=2(2^0-1)+1=2^1-1. By simplifying the original equation we get F[n]=2^n-1, which is the algorithm for the minimum number of moves to solve the Tower of Hanoi. It is linked to Singapore's ports as they both require stacking and unstacking, and there are restrictions imposed in both cases. Hence we used dynamic programming to solve the different case scenarios, and we had the following conclusions.

When there are fewer containers, less reshuffling is required. When there are 2 stacking slots, only 4 moves are needed to stack 2 containers and unload them, regardless of their entering and leaving sequence.

When there are 3 stacking slots with 3 containers, the minimum number of moves to stack and unload the containers is 6 moves. The only exception is when the entering and leaving sequence are the same (i.e. Entering sequence: Yellow, Blue, Red; Leaving sequence: Yellow, Blue, Red). In this case, the minimum number of moves is 7 moves as more moves are needed to make the yellow and the blue containers at the top of any stack to unload them.

For situations where the number of containers is similar to the number of stacking slots, no reshuffling will be needed with strategic stacking of containers.

The most efficient method of stacking would be to let the container which has to be unloaded first to not be stacked below any other containers.

Limitations

The Case Scenarios in our research are much smaller scale than those encountered at the ports in real life. It would be too difficult and complex to consider other constraints and factors such as weight, maximum stacking height and destination port.

Further research

One can consider 4, 5 or more containers to find an algorithm that can find the most efficient way of stacking the containers, consider other factors and constraints such as weight and loading sequence, or consider having 3 or more stacking slots.

References

1. Arpit Mishra (2016, December 26). Tower of Hanoi recursion game algorithm explained. Retrieved on: 6 February 2021. From HackerEarth Blog website: https://www.hackerearth.com/blog/developers/tower-hanoi-recursion-game-algorithm-explained/#:~:text=Tower%20of%20Hanoi%20consists%20of

2. Kim, K., & Kim, H. (2002). The optimal sizing of the storage space and handling facilities for import containers. *Transportation Research Part B: Methodological*, *36*(9), 821-835. doi: 10.1016/s0191-2615(01)00033-9

3. Mi, W., Liu, Y., Xia, M., Shen, Y., & Zhao, N. (n.d.). *Digital Management of Container Terminal Operations*.

4. Taleb-Ibrahimi, M., de Castilho, B., & Daganzo, C. (1993). Storage space vs handling work in container terminals. *Transportation Research Part B: Methodological*, *27*(1), 13-32. doi: 10.1016/0191-2615(93)90009-y

5. Towers of Hanoi. (2019). Towers of Hanoi. Retrieved on: 6 February 2021. From Khan Academy website: https://www.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoin

6. Alyssa Dalton (2021). Retrieved 9 August 2021, from http://mercury.webster.edu/aleshunas/Support%20Materials/Presentations/Dalton_Dynamic%20Programming2.pdf

7. (2021). Retrieved 15 August 2021, from http://civil.colorado.edu/~balajir/CVEN5393/lectures/chapter-10.pdf