Mission Impossible 7 - 2h 42m
Director: Christopher McQuarrie
Plot: “A submarine AI has gone wrong and is planning to end the world. Spies from other nations attempt to stop it. Part 1 of 2 Part Movie”
Year Released: 2023
Rating: PG-13

No Time to Die: 2h 43m
Director: Cary Joji Fukunaga
Plot: “Former villain has a plan to kill everyone on the planet with a virus. James Bond attempts to stop him, but sacrifices himself.
Year Released: 2021
Rating: PG-13

Fast and Furious: Tokyo Drift - 1h 44m
Director: Justin Lin
Plot: “sigma male learns how to drift and gets all the girls.”
Year Released: 2006
Rating: PG-13

Hitman: Agent 47 - 1h 36m
Director: Aleksander Bach
Plot: “A hitman attempts to kill his target, but faces some challenges.”
Year Released: 2015
Rating: R

Uncharted - 1h 56m
Director: Ruben Fleischer
Plot: “Younger brother attempts to find older brother, but gets caught up in a treasure hunt.”
Year Released: 2022
Rating: PG-13

Bahubali: The Beginning - 2h 39
Director: S. S. Rajamouli
Plot: “Man climbs over a mountain and tries to save a princess, who is his mom, and learns about his past.”
Year Released: 2015
Rating: PG

Bahubali: The Conclusion - 3h 17m
Director: S. S. Rajamouli
Plot: “Man that climbed over a mountain past is happening, and then back in the present tries to help defeat the bad guys from his past.”
Year Released: 2017
Rating: PG

Total: 16hr 37m

Niece
Oz the Great and the Powerful 2hrs 10mins
Director: Sam Raimi
Plot: “A man from the circus goes away from Kansas, and figures out that he is the Wizard of Oz. He must figure out the good from the bad.”
Year Released: 2013
Rating: PG

The Karate Kid 2hr 2mins
Director: John G. Avildsen
Plot: “Kid gets bullied and enters a karate competition. He finds a sensei to win the competition.”
Year Released: 1984
Rating: PG

Fantastic Mr Fox 1hr 27mins
Director: Wes Anderson
Plot: “A human fox breaks a promise and raids his neighbor. He starts going back to primal instincts and goes underground. Mr. Fox has to face opposition.”
Year Released: 2009
Rating: PG

Total: 5h 39min

Design Plan

Movie Class

- Most of movie class is already created just added new instance variable String rating
- Created getters and setters for instance variable rating

Node Class

- Create instance variables Object data
- Create instance variable Node nextNode
- Create instance variable Movie highestRuntime (This will be used in the head node only)
- Create instance variable Movie lowestRuntime (This will also be used in head only)
- Create Getters and Setters for all instance variables data, nextNode, highestRuntime, LowestRuntime
- Constructor Node
    - Get data parameter and next Node parameter
    - Set data and node parameter to proper variable
    - Set highest and lowest runtimes to null

LinkedListMovie Class

- Create instance variable head
- Constructor set head equal to null

- boolean isEmpty
    - Return true or false based on head == null

- Object getHeadNode
    - Return head

- Void setHeadNode (Node node)
    - Check if list is empty
            - If is then make movie object and set equal to node.getData
        - Call newHeadNode then return
    - Make movie objects for highestMovie, lowestMovie, and given Node movie
    - Check to see if given movie is higher or lower than highestMovie or lowestMovie
    - If is then set the corresponding movie to node movie
    - Make node.setNext(head.getNext) (head node gets removed)
    - Make head = node
    - setHighestRuntime(highest)
    - setLowestRuntime(lowest)

- Void newHeadNode (Object obj)
    - Create node with given object and make next = null
    - Make movie object equal to node.getData
    - Check to see if list is empty if is then

```
        - Make head = node
        - setHighest and lowest Runtimes = movie object
    - else
        - Create current movie = node.getdata
        - Create highest and lowest movie by calling head.gethighest/lowest
        - Set head = node
        - Check to see if Check to see if given movie is higher or lower than highestMovie
          or lowestMovie
        - If is then set the corresponding movie to node movie
        - Else
            - set highest and lowest to already highest and lowest

- Void addNewTail (Object obj)
    - Make Node current = head
    - Make Node previous = null
    - Make while loop until current != null
    - Make previous = current
    - And make current= current.getNext
    - Outside while
    - Check to see if previous node is null (list is empty
    - Call newHeadNode(obj)
    - else
        - Make new node with obj and next equal to null
        - Create current movie = node.getdata
        - Create highest and lowest movie by calling head.gethighest/lowest
        - Set head = node
        - Check to see if Check to see if given movie is higher or lower than highestMovie
          or lowestMovie
        - If is then set the corresponding movie to node movie
        - Else
            - set highest and lowest to already highest and lowest
    - Set previousNode.setNext equal to n

- Void addNewNode (Object object, int nodeLocation)
    - Check to see if list is empty
        - If is then call newHeadNode (object)
    - Make counter num
    - Make previous node = null
    - Make current node = head
    - While loop through currentNode.getNext != null && counter < nodeLocation
    - Previous = current
    - Current = current.getNext
    - Counter++
    - Outside while
```

```
        -   Make new node object with object parameter
        -   Create current movie = node.getdata
        -   Create highest and lowest movie by calling head.gethighest/lowest
        -   Check to see if Check to see if given movie is higher or lower than highestMovie or
            lowestMovie
        -   If is then set the corresponding movie to node movie
        -   Else
                -   set highest and lowest to already highest and lowest
        -   Set previous.setnext = n
        -   currentNode.setNext(currentNode.getNext())

-   Object removeHead ()
        -   Make temp node = head
        -   Create current movie = node.getdata
        -   Create highest and lowest movie by calling head.gethighest/lowest
        -   Check to see if current.getMinutesInMintues == highest or lowest time
                -   Make current node = head.getNext
                -   Highest = current.getData
                -   Lowest = current.getData
                -   While loop until current == null
                        -   Current movie = currentNode.getData
                        -   Check to see if Check to see if given movie is higher or lower than
                            highestMovie or lowestMovie
                        -   If is then set the corresponding movie to node movie
                        -   Current = current.getNext
        -   Head = head.getNext
        -   Head.sethighest and lowest = corresponding movie
        -   Return temp.getData

-   Object removeFromTail ()
        -   If is empty
                -   Throw exception cannot remove from empty list
        -   Make current node = head
        -   Make previou node = null
        -   While current node.getNext != null
                -   Previous = current
                -   Current = current.getNext
        -   Object removed = current.getData
        -   If previous == null
                -   Head == null
        -   Else
                -   Create current movie = node.getdata
                -   Create highest and lowest movie by calling head.gethighest/lowest
                -   Check to see if current.getMinutesInMintues == highest or lowest time
                        - Make current node = head.getNext
                        - Highest = current.getData
                        - Lowest = current.getData
                        - While loop until current == null
                            - Current movie = currentNode.getData
                            - Check to see if Check to see if given movie is higher or lower than
                              highestMovie or lowestMovie
                            - If is then set the corresponding movie to node movie
                            - Current = current.getNext
                            - Previous.setNext == null
                        - Head.sethighest and lowest = corresponding movie
        - Return Removed

- String sortMovies (int sortType)
        - Create LinkListMovie original = new LinkListMovie
        - Create LinkListMovie sorted = new LinkListMovie
        - Make current node = head
        - Make int movieAmount = 0
        - While current != null
                - Make Movie object temp = current.getData
                - Add to original list
                - Current = current.getNext
                - movieAmount++
        - Check to see what type of sort
                - For loop through i < movieAmount
                        - Create Node currentNode = original.getHead
                        - Create Movie highestMovie = currentNode.getDada
                        - Make counter int = 0
                        - Make deleteIndex int = 0
                        - Make node for insideLoop = original.getHead
                            - While insideLoop != null
                            - Movie currentMovie = insideLoop.getData
                            - Check for highest title, director, year
                            - If is then
                                - Set highestMovie = currentMovie
                                - deleteIndex = counter
                            - Counter++
                            - insideLoop = insideLoop.getNext
                        - Add highestMovie to end of sorted class
                        - Delete node at deleteIndex
        - Return sorted.toString

- Private void deleteNode (int index) (helper method)
        - Node currentNode equal to head
```

```
        -   Make Node previousNode = null
        -   Int counter = 0;
        -   If index is 0
            -   Make movie object for highest, lowest, and current movie
            -   Current = head.getNext.getData
            -   Make a node tempNode= head.getNext
            -   Check to see if higher != null && lower not equal to null
                -   While tempNode != null
                    -   Check to see if Check to see if given movie is higher or lower than highestMovie or lowestMovie
                    -   If is then set the corresponding movie to node movie
                    -   tempNode = tempNode.getNext
            -   Sethighest and lowestMovies to highest and lowest
            -   Head = current.getNext
            -   Return
        -   While counter < index
        -   PreviousNode = current
        -   Current = current.getNext
        -   Counter++
        -   previousNode.setNext = currentNode.setNext
        -   Make movie object for highest, lowest, and current movie
        -   Current = head.getNext.getData
        -   Make a node tempNode= head.getNext
        -   Check to see if higher != null && lower not equal to null
            -   While tempNode != null
                -   Check to see if Check to see if given movie is higher or lower than highestMovie or lowestMovie
                -   If is then set the corresponding movie to node movie
                -   tempNode = tempNode.getNext
        -   Sethighest and lowestMovies to highest and lowest

    -   String totalRuntime ()
        -   Node current = head
        -   Int time = 0;
        -   While current != null
            -   Make movie object on currentNode
            -   Time += movie object time
            -   Current = current.getNext
        -   Int hrs = time / 60
        -   Int min = time % 60
        -   Return hrs and min in given order

    -   String moviesOfRating ()
        -   Make string builder
        - Check to see which rating is wanted
        - Node current = head
        - While current != null
            - Make movie object of current
            - Check for rating
            - If is then append to stringbuilder
        - Current = current.getNext
    - Return stringbuilder

    - String findPlot (String title)
        - Make Node current and temp movie object
        - While current != null
            - Make movie object from current node
            - Check to see if titles match with movie
            - If is then set temp equal to movie and break
            - Current = current.getNext
        - If movie1 != null return temp.getPlot
        - Return null

    - String getMaxAndMinRun
        - Return highest + head.getHighest + lowest + head.getLowest

    - String toString
        - Make string builder
        - Make current node = head
        - While current != null
            - Append current.getData
            - Current = current.getNext
        - Return stringbuilder
```

AirShip Class

```
    - Make to linklist one for niece one for nightsong
    - Create all movie objects and set all movie object into linklist
    - Boolean stay = true
    - While stay
        - Display choices and ask user what they want
        - Store ans
        - If ans 0
            - Stay = false
        - If ans 1
            - As which movie list they want
            - Return proper movie list asked for
        - If ans 2
            - Ask for movie title
        -   Call findPlot (movieTitle)
        -   Check to see if return is null if is then print could not find if not print the the movie
            plot
    -   If ans 3
        -   Ask for which list runtime
        -   Display proper runtime by calling proper class
    -   If ans 4
        -   Ask the user how they would like to print it
            -   Then ask the user which list they want
            -   One the other or both
            -   If both then add the movies from niece to nightsong list then call sort then
                remove the movies from nightsong
    -   If ans 5
        -   Ask for which rating
        -   Add the movies from niece to night song and call the proper method with proper
            parameter then remove the movies from nightsong
    -   If 6
        -   Add movies to nightsong again
        -   Then call the max and min runtime
        -   Remove the movies from nightsong
```