Selection sort

```
private QueueTAD<Game> selectionSort(ArrayList<Game> list){
        QueueTAD<Game> queue = new QueueTAD<>();                                    1

        for (int i = 0; i < list.size(); i++) {                                     n+1
            Game minor = list.get(i);                                               n
            int pos = i;                                                            n

            for (int j = i + 1; j < list.size(); j++) {                             n(n−1)/2 + n
                if (minor.compareTo(list.get(j)) > 0) {                             n(n−1)/2
                    minor = list.get(j);                                            n(n−1)/2
                    pos = j;                                                        n(n−1)/2
                }
            }
            Game temp = list.get(i);                                                n
            list.set(i, minor);                                                     n
            list.set(pos, temp);                                                    n
        }

        for (Game game : list) {                                                    n+1
            queue.add(game);                                                        n
        }

        return queue;                                                               1
    }
```

$$C_1 + C_2 + C_{12} + C_{14} + C_3 n + C_4 n + C_5 n + C_5\left(\frac{n(n-1)}{2}\right) + C_6\frac{n(n-1)}{2} + C_7\frac{n(n-1)}{2} + C_8\frac{(n(n-1))}{2} + C_9 n + C_{10} n + C_{11} n + C_{12} n + C_{13} n$$

$$C_1 + C_2 + C_{12} + C_{14} + n(C_3 + C_4 + C_5 + C_9 + C_{10} + C_{11} + C_{12} + C_{13}) + \frac{n(n-1)}{2}(C_5 + C_6 + C_7 + C_8)$$

$$A + Bn + C\left(\frac{n(n-1)}{2}\right)$$

$$O(n^2)$$

Insertion sort:

```
private QueueTAD<Game> insertionSort(ArrayList<Game> list){

        QueueTAD<Game> queue = new QueueTAD<>();                    1

        for (int i = 0; i < list.size(); i++) {                     n+1
            Game minor = list.get(i);                               n
            for (int j = i + 1; (j < list.size()); j++) {           n(n-1)/2 + n
                if (minor.compareTo(list.get(j)) > 0) {             n(n-1)/2
                    Game temp = list.get(j);                        n(n-1)/2
                    list.set(j, minor);                             n(n-1)/2
                    list.set(i, temp);                              n(n-1)/2
                }
            }
        }

        for (Game game : list) {                                    n+1
            queue.add(game);                                        n
        }

        return queue;                                               1
    }
```

$$C_1 + C_9 + C_{11} + C_2 n + C_3 n + C_4 n + C_9 n + C_{10} n + C_4 \frac{n(n-1)}{2} + C_5 \frac{n(n-1)}{2} + C_6 \frac{n(n-1)}{2} + C_7 \frac{n(n-1)}{2} + C_8 \frac{n(n-1)}{2}$$

$$C_1 + C_9 + C_{11} + n(C_2 + C_3 + C_4 + C_9 + C_{10}) + \frac{n(n-1)}{2}(C_4 + C_5 + C_6 + C_7 + C_8)$$

$$A + Bn + C\left(\frac{n(n-1)}{2}\right)$$

$$O(n^2$$