Complexity

Temporal

|  |  |  |
| --- | --- | --- |
| 1 | File file = new File(txt); | 1 |
| 2 | hashTable = new HashTable<>(count); | 1 |
| 3 | FileInputStream fis = new FileInputStream(file); | 1 |
| 4 | BufferedReader reader = new BufferedReader(new InputStreamReader(fis)); | 1 |
| 5 | String line; | 1 |
| 6 | Try{ | 1 |
| 7 | while ((line = reader.readLine()) != null) | n+1 |
| 8 | String[] parts = line.split(" "); | N |
| 9 | Passenger p = new Passenger(parts[0],parts[1],Boolean.parseBoolean(parts[2]),Integer.parseInt(parts[3]),Integer.parseInt(parts[4]), Integer.parseInt(parts[5])); | N |
| 10 | int fila = (Integer.parseInt(parts[3]) - 1) / plane.getColumn() + 1; | N |
| 11 | int columna = (Integer.parseInt(parts[3]) - 1) % plane.getColumn() + 1; | N |
| 12 | p.setSeat(new Seat(columna, fila)); | N |
| 13 | int p1Dist = Math.min(p.getSeat().getColumn(), plane.getColumn() - p.getSeat().getColumn() + 1); | N |
| 14 | p.setDis(p1Dist); | N |
| 15 | hashTable.insert(parts[5], p); | N |
| 16 | fis.close(); | 1 |
| 17 | catch (FileNotFoundException e) | 1 |
| 18 | e.printStackTrace(); | 1 |
| 19 | catch (IOException e) | 1 |
| 20 | e.printStackTrace(); | 1 |
| 21 |  |  |
| 21 |  |  |

Total: 12+9n

Notation big O

g(n) = 9n +12 => O(n^2)

|  |  |  |
| --- | --- | --- |
| 1 | int index = hash(key); | 1 |
| 2 | NodeHash<K, V> current = list[index]; | 1 |
| 3 | while (current != null) | N+1 |
| 4 | if (current.getKey().equals(key)) | N |
| 5 | if (current.getPrev() == null) | N |
| 6 | list[index] = current.getNext(); | N |
| 7 | current.getPrev().setNext(current.getNext()); | N |
| 8 | if (current.getNext() != null) | N |
| 9 | current.getNext().setPrev(current.getPrev()); | N |
| 10 | return | 1 |
| 11 | current = current.getNext(); | 1 |

Total=5+7n

Notion big O

g(n) = 5+7n => O(n^2)

space:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | File file = new File(txt); | 8 bytes | 1 |
| 2 | hashTable = new HashTable<>(count); | 32 bytes | 1 |
| 3 | FileInputStream fis = new FileInputStream(file); | 8 KB:8192bytes | 1 |
| 4 | BufferedReader reader = new BufferedReader(new InputStreamReader(fis)); | 8 KB:8192bytes | 1 |
| 5 | String line; | 16 bytes | 1 |
| 6 | Try{ |  | 1 |
| 7 | while ((line = reader.readLine()) != null) |  | n+1 |
| 8 | String[] parts = line.split(" "); | 16 bytes\*n | N |
| 9 | Passenger p = new Passenger(parts[0],parts[1],Boolean.parseBoolean(parts[2]),Integer.parseInt(parts[3]),Integer.parseInt(parts[4]), Integer.parseInt(parts[5])); | 32 bytes | N |
| 10 | int fila = (Integer.parseInt(parts[3]) - 1) / plane.getColumn() + 1; | 4 bytes | N |
| 11 | int columna = (Integer.parseInt(parts[3]) - 1) % plane.getColumn() + 1; | 4 bytes | N |
| 12 | p.setSeat(new Seat(columna, fila)); | 32 bytes | N |
| 13 | int p1Dist = Math.min(p.getSeat().getColumn(), plane.getColumn() - p.getSeat().getColumn() + 1); | 4 bytes | N |
| 14 | p.setDis(p1Dist); |  | N |
| 15 | hashTable.insert(parts[5], p); | 32 bytes | N |
| 16 | fis.close(); |  | 1 |
| 17 | catch (FileNotFoundException e) |  | 1 |
| 18 | e.printStackTrace(); |  | 1 |
| 19 | catch (IOException e) |  | 1 |
| 20 | e.printStackTrace(); |  | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | int index = hash(key); | 4 bytes | 1 |
| 2 | NodeHash<K, V> current = list[index]; | 32 bytes | 1 |
| 3 | while (current != null) |  | N+1 |
| 4 | if (current.getKey().equals(key)) |  | N |
| 5 | if (current.getPrev() == null) |  | N |
| 6 | list[index] = current.getNext(); | 32 bytes\*n | N |
| 7 | current.getPrev().setNext(current.getNext()); |  | N |
| 8 | if (current.getNext() != null) |  | N |
| 9 | current.getNext().setPrev(current.getPrev()); |  | N |
| 10 | return |  | 1 |
| 11 | current = current.getNext(); | 32 bytes | 1 |