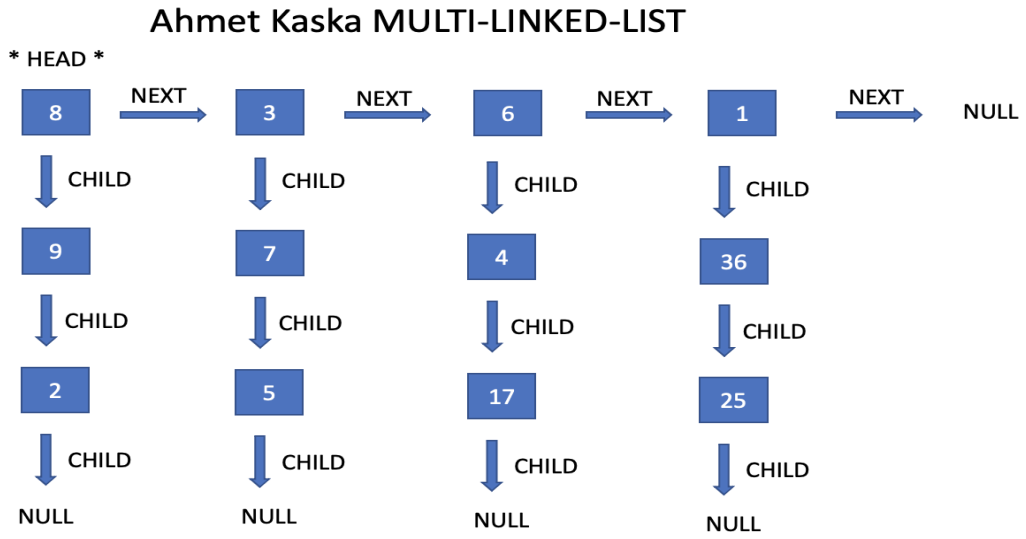
DATA STRUCTURES WORK 1 REPORT

Proje analizi sonucunda oluşturulan tasarımım.



Projede bir adet çoklu bağlı liste ile 2048 oyunu yapmamız istendi. Yapı olarak baş düğüm ve baş düşümün sonrasını ebeveyin düğüm olarak tanımlandırıldı. Düğüm sınıfının yapıcı methodunda posizyon değeri istenildi. Pozisyon bilgisi tutularak ebeveyin düğümler konumlandırıldı. Pozisyon degerinin yapıcı methodta istenmesinin sebebi boş düğümlerin çoklu bağlı listemizde bulundurulmaması içindir. Bu işlemler AddParentNode(int position) fonksiyonu ile gerçekleştirildi.

Ebeveyin düğümlere, çocuk düğüm ekleyebilmek için findIndex(int index), insertChildNode(int parentIndex, int newValue) ve fonksiyonları kullanıldı. findIndex methodu parametre olarak aldığı index değeri ile pozisyon bilgisi aynı olan düğümü döndürdü. Böylece ebeyin düğüme eriştik. Sonrasında ebeyin düğümün, çocuk düğümleri üzerinde gezinildi ve boş olan alana yeni düğüm eklendi.

Böylece yapımız oluştu. 2048 oyununun tam işlevini görebilmesi için AddTwoNodesWithSameData(int position) fonksiyonu kullanıldı. Parametre olarak alınan pozisyon bilgisi ile ebeveyin düğüme ulaştık. Ebeveyin düğümün şuanı ve öncesi adında iki düğüm ile üzerinde gezinildi. Eğer şuan düğüm ile öncesi düğümün verileri aynıysa AddTwoNodesWithSameData ile toplama işlemi yapıldı ve öncesi düğümün verisi toplam değeri ile güncellendi.

Output

```
2 --> Null
|
Null
****************************************************************
Node added. Position of new node is 4. Data of new node is 2.
2 --> Null
|
2 --> Null
|
Null
****************************************************************
Node added. Position of new node is 2. Data of new node is 4.
2 --> Null
|
4 --> Null
|
2 --> Null
|
Null
****************************************************************
Node added. Position of new node is 3. Data of new node is 2.
2 --> Null
|
4 --> Null
|
2 --> Null
|
2 --> Null
|
Null
****************************************************************
Node added. Position of new node is 5. Data of new node is 4.
2 --> Null
|
4 --> Null
|
2 --> Null
|
2 --> Null
|
4 --> Null
|
Null
****************************************************************
Child node added. Position of new child node is 2. Data of new child node is 2.
2 --> Null
|
4 --> 2 --> Null
|
2 --> Null
|
2 --> Null
|
4 --> Null
|
Null
****************************************************************
```

```
Child node added. Position of new child node is 5. Data of new child node is 4.
2 --> Null
|
4 --> 2 --> Null
|
2 --> Null
|
2 --> Null
|
4 --> 4 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 5. Data of updated node is 8.
2 --> Null
|
4 --> 2 --> Null
|
2 --> Null
|
2 --> Null
|
8 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 1. Data of new child node is 8.
2 --> 8 --> Null
|
4 --> 2 --> Null
|
2 --> Null
|
2 --> Null
|
8 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 1. Data of new child node is 8.
2 --> 8 --> 8 --> Null
|
4 --> 2 --> Null
|
2 --> Null
|
2 --> Null
|
8 --> Null
|
Null
************************************************************
```

```
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 1. Data of updated node is 16.
2 --> 16 --> Null
|
4 --> 2 --> Null
|
2 --> Null
|
2 --> Null
|
8 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 2. Data of new child node is 32.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> Null
|
2 --> Null
|
2 --> Null
|
8 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 3. Data of new child node is 2.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> Null
|
2 --> 2 --> Null
|
2 --> Null
|
8 --> Null
|
Null
**************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 4.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> Null
|
4 --> Null
|
2 --> Null
|
8 --> Null
|
Null
**************************************************************
```

```
Child node added. Position of new child node is 3. Data of new child node is 64.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> Null
|
4 --> 64 --> Null
|
2 --> Null
|
8 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 4. Data of new child node is 16.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> Null
|
4 --> 64 --> Null
|
2 --> 16 --> Null
|
8 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 2. Data of new child node is 64.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> Null
|
2 --> 16 --> Null
|
8 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 3. Data of new child node is 32.
2 --> 16 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> 32 --> Null
|
2 --> 16 --> Null
|
8 --> Null
|
Null
**************************************************************
```

```
Child node added. Position of new child node is 1. Data of new child node is 16.
2 --> 16 --> 16 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> 32 --> Null
|
2 --> 16 --> Null
|
8 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 1. Data of updated node is 32.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> 32 --> Null
|
2 --> 16 --> Null
|
8 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 5. Data of new child node is 16.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> 32 --> Null
|
2 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 3. Data of new child node is 32.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> 32 --> 32 --> Null
|
2 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
```

```
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 64.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 64 --> 64 --> Null
|
2 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 128.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 64 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 2. Data of new child node is 64.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 64 --> 64 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 2. Data of updated node is 128.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
```

```
Child node added. Position of new child node is 4. Data of new child node is 8.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 8 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 4. Data of new child node is 4.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 8 --> 4 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 4. Data of new child node is 2.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 8 --> 4 --> 2 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 4. Data of new child node is 2.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 8 --> 4 --> 2 --> 2 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
```

```
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 4. Data of updated node is 4.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 8 --> 4 --> 4 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 4. Data of updated node is 8.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 8 --> 8 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 4. Data of updated node is 16.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 16 --> 16 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
```

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 4. Data of updated node is 32.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> Null
|
4 --> 128 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
Child node added. Position of new child node is 2. Data of new child node is 2.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
Child node added. Position of new child node is 3. Data of new child node is 64.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
Child node added. Position of new child node is 3. Data of new child node is 32.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 32 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
```

```
Child node added. Position of new child node is 3. Data of new child node is 16.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 32 --> 16 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 3. Data of new child node is 8.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 32 --> 16 --> 8 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Child node added. Position of new child node is 3. Data of new child node is 8.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 32 --> 16 --> 8 --> 8 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 16.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 32 --> 16 --> 16 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
************************************************************
```

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 32.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 32 --> 32 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 64.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 64 --> 64 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 128.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 128 --> 128 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
****************************************************************
```

```
**************************************************************
Since the data of the child node and the previous node are equal,
the addition process was performed and the data of the previous node was updated.
Position of updated node is 3. Data of updated node is 256.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> Null
|
4 --> 256 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 2. Data of new child node is 4.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> 4 --> Null
|
4 --> 256 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
**************************************************************
Child node added. Position of new child node is 2. Data of new child node is 8.
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> 4 --> 8 --> Null
|
4 --> 256 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
**************************************************************
Game Over!
2 --> 32 --> Null
|
4 --> 2 --> 32 --> 128 --> 2 --> 4 --> 8 --> Null
|
4 --> 256 --> Null
|
2 --> 32 --> Null
|
8 --> 16 --> Null
|
Null
**************************************************************
BUILD SUCCESSFUL (total time: 0 seconds)
```