

VERİ YAPILARI 3. ÖDEVİ

DERSİN ADI : VERİ YAPILARI (BLM 2512)

ÖĞRENCİNİN ADI : YUNUS

ÖĞRENCİNİN SOYADI : KARATEPE

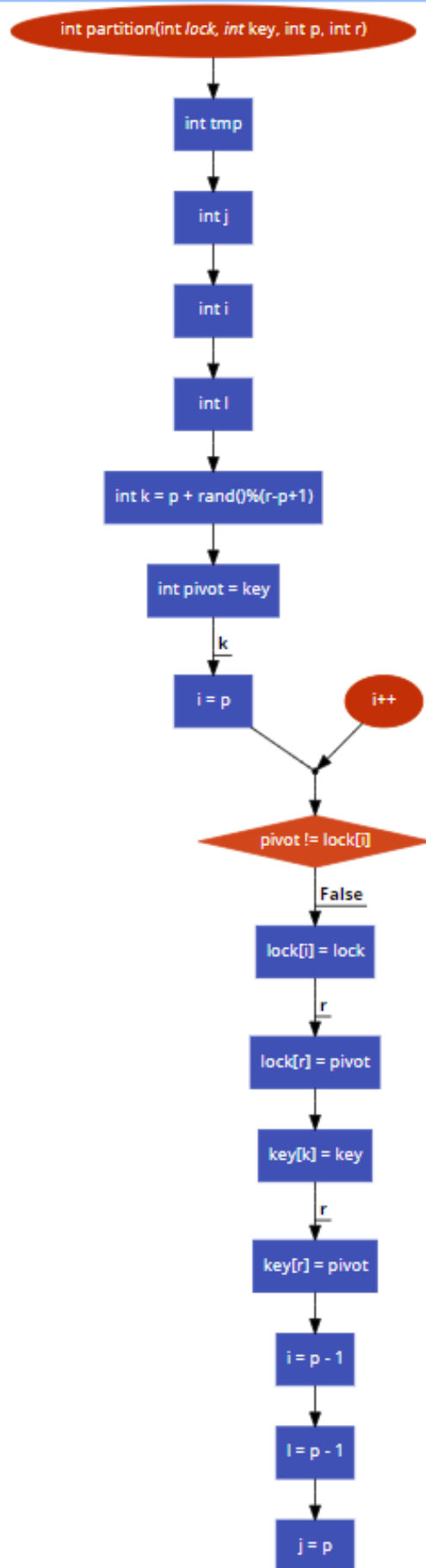
ÖĞRENCİNİN NUMARASI : 17011051

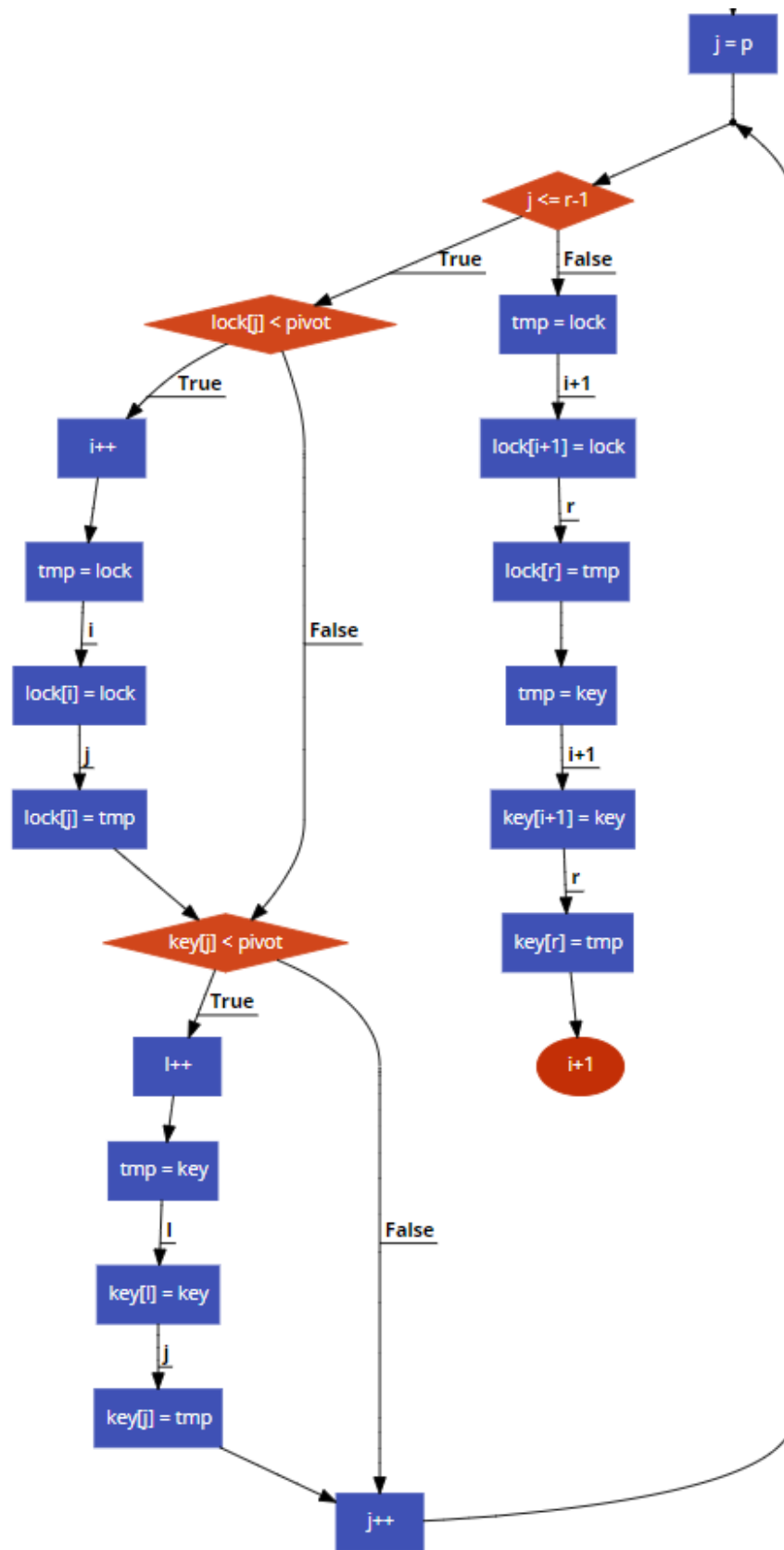
ÖDEV KONUSU : BÖL VE YÖNET ALGORİTMALARI

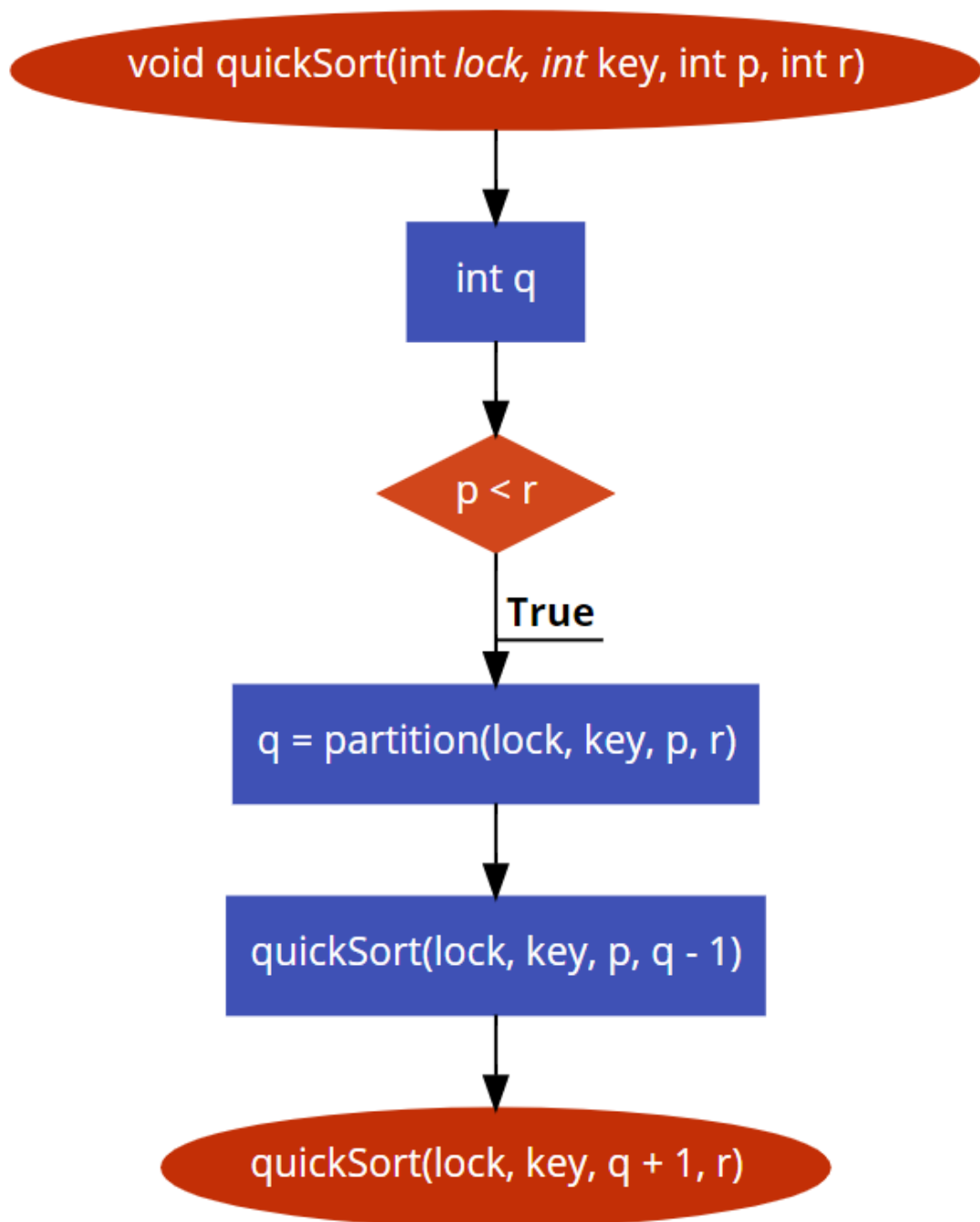
BÖLÜM 1 : YÖNTEM

ALGORİTMANIN KISACA MANTIĞI :

Yazdığım algoritma quickSort sıralama algoritmasını tamamen kullanıyor. Bunun yanı sıra pivotumuzu random seçmemiz gerekiyor. Benim yazdığım quickSort algoritmasında pivot random olarak “key” dizisinden seçiliyor. Daha sonra “lock” dizisindeki her elemanla bu seçilen pivot karşılaştırılıp eşit olana kadar devam edene yani bu “key” değerinin “lock” daki değerini bulana kadar dönüyor. Bulunduktan sonra “lock” da bulunan bu değer ile “lock” dizisinin son elemanı her değiştiriliyor. “key” dizisinde de aynı işlem “key” deki pivot için de yapılıyor. Yani her iki dizide de pivot sona alınmış ve sondaki eleman, o dizinin kendi pivotunun indisi neyse, oraya yerleştirilmiş oluyor. Daha sonrasında yapmamız gereken tek şey quickSortun klasik “partition” algoritmasını kullanmak. 2 adet indis oluyor. Bunlardan birisi dizinin başından sonuna kadar giden, diğeri ise ilk indisteki eleman pivottan küçük ise artan iki indis. En sonunda şartlı olarak artan indis ile pivotun yeri değiştiriliyor. Bu her iki dizi için de yapılıyor. Dizilerdeki elemanlar aynı olduğundan dolayı toplamda pivottan küçük olan eleman sayısı da aynı. Sonuç olarak pivot her iki dizide de aynı noktaya gelmiş oluyor. Böylece “key” – “lock” eşlenmiş oluyor ve pivotun yerini quickSort fonksiyonuna döndürüyor. Bütün bu algoritmayı her dönen pivotun solu ve sağ için de ($sol < sağ$ olduğu sürece) yaptığımız zaman iki dizideki bütün elemanlar küçükten büyüğe sıralanmış oluyor.







BÖLÜM 2 : UYGULAMA

```
C:\Users\yunus\OneDrive\Masaüstü\Veri Yapıları\dev 3\main.exe
Boyutu giriniz : 10
1. anahtar : 10
2. anahtar : 9
3. anahtar : 8
4. anahtar : 7
5. anahtar : 6
6. anahtar : 5
7. anahtar : 4
8. anahtar : 3
9. anahtar : 2
10. anahtar : 1

Kapılar   : 7 6 1 5 10 2 8 4 3 9
Anahtarlar : 10 9 8 7 6 5 4 3 2 1
-----
Kapılar   : 1 2 3 4 5 6 7 8 9 10
Anahtarlar : 1 2 3 4 5 6 7 8 9 10
-----
Process exited after 10.55 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\yunus\OneDrive\Masaüstü\Veri Yapıları\dev 3\main.exe
Boyutu giriniz : 10
1. anahtar : 1
2. anahtar : 2
3. anahtar : 3
4. anahtar : 4
5. anahtar : 5
6. anahtar : 6
7. anahtar : 7
8. anahtar : 8
9. anahtar : 9
10. anahtar : 10

Kapılar   : 8 4 3 1 2 6 9 5 10 7
Anahtarlar : 1 2 3 4 5 6 7 8 9 10
-----
Kapılar   : 1 2 3 4 5 6 7 8 9 10
Anahtarlar : 1 2 3 4 5 6 7 8 9 10
-----
Process exited after 12.46 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\yunus\OneDrive\Masaüstü\Veri Yapıları\dev 3\main.exe
Boyutu giriniz : 10
1. anahtar : 32
2. anahtar : 64
3. anahtar : 73
4. anahtar : 46
5. anahtar : 32
6. anahtar : 78
7. anahtar : 46
8. anahtar : 78
9. anahtar : 90
10. anahtar : 53

Kapılar   : 64 73 32 78 53 90 46 46 78 32
Anahtarlar : 32 64 73 46 32 78 46 78 90 53
-----
Kapılar   : 32 32 46 46 53 64 73 78 78 90
Anahtarlar : 32 32 46 46 53 64 73 78 78 90

-----
Process exited after 9.528 seconds with return value 0
Press any key to continue . . .
```

BÖLÜM 3 : SONUÇ

ALGORİTMANIN KARMAŞIKLIĞI =

Her seferinde pivotun sağı için ve solu için n lik bir dizinin hepsini geziyor. Bu da :

$$= (\log_2 N) * N \quad \text{olur.}$$