李青航 SA22225226

14.3-2

Algorithm 1 INTERVAL-SEARCH(T, i)

```
$x = T.root$
**while** $x \neq T.nil$ and $i$ does not overlap $x.int$ **do**
    **if** $x.left \neq T.nil$ and $x.left.max > i.low$ **then**
        $x = x.left$
    **else**
        $x = x.right$
    **end if**
**end while**
**return** $x$
```

15.1-4

见算法2，3. 数组s保存,$s[i]$表示长度为i的钢条，从哪里切

Algorithm 2 MEMOIZED-CUT-ROD(p, n)

```
let $r[0...n]$ be a new array
let $s[0...n]$ be a new array
**for** $i = 0$ to $n$ **do**
    $r[i] = -\infty$
    $s[i] = 0$
**end for**
**return** MEMOIZED-CUT-ROD-AUX($p, n, r, s$)
```

15.2-1

方案 $(A_1A_2)((A_3A_4)(A_5A_6))$

最小代价$5 \cdot 50 \cdot 6 + 3 \cdot 12 \cdot 5 + 5 \cdot 10 \cdot 3 + 3 \cdot 5 \cdot 6 + 5 \cdot 3 \cdot 6 = 1500 + 180 + 150 + 90 + 90 = 2010$

15.2-2

见算法4

Algorithm 3 MEMOIZED-CUT-ROD-AUX(p, n, r, s)

```
if $r[n] \geq 0$ then
    return $r[n]$
end if
if $n == 0$ then
    $q = 0$
else if $q = -\infty$ then
    for $i = 1$ to $n$ do
        $t = p[i]$+MEMOIZED-CUT-ROD-AUX$(p, n - i, r, s).q$
        if $t \geq q$ then
            $q = t$
            $s[n] = i$
        end if
    end for
end if
$r[n] = q$
return $q$ and $s$
```

Algorithm 4 MATRIX-CHAIN-MULTIPLY(A, s, i, j)

```
if $i == j$ then
    return $A_i$
end if
return MATRIX-CHAIN-MULTIPLY$(A, s, i, s[i, j])\cdot$ MATRIX-CHAIN-MULTIPLY$(A, s, s[i, j] + 1, j)$
```

15.3-1

穷举快

穷举法根据组合数学知识，时间复杂度是一个指数复杂度。而RECURSIVE-MATRIX-CHAIN也是一个指数复杂度的，但是还有递归函数调用开销，还会重叠算某些子问题（穷举没有重叠）,重复调用函数。

15.3-2

图太难画，略。归并排序并没有重叠子问题，每次归并，没有重复可用的操作，所以备忘技术没有用