DSAA Ass 11 12310401 王子恒

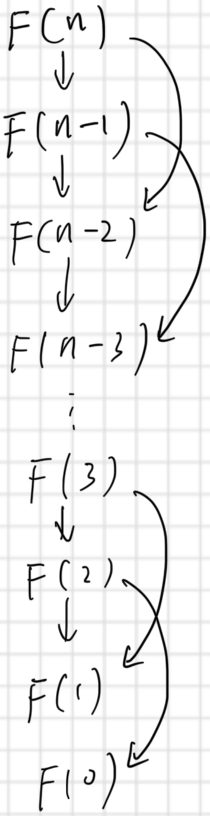
1.
```
Bottom-Up-Cut-Rod(p.n)
Let r[0…n] be a new array
r[0]=0
for j=1 to n do
    q=-∞
    for i=1 to j do
        q=max(q, p[i]+r[j-i]-c)
    r[j]=q
return r[n]
```

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
        r[i] = -∞
        s[i] = 0
    return MEMOIZED-CUT-AUX(p,n,r,s)

MEMOIZED-CUT-ROD-AUX(p,n,r,s)
 if r[n] ≥0
     return r[n],s[n]
 if n==0
     q=0
 else q=-∞
 cut_position=0
 for i=1 to n
     tmp = p[i]+MEMOIZED-CUT-ROD-AUX(p,n-i,r)[0]
     if tmp > q
         q= tmp
         cut_position=i
 r[n]=q
 s[n]= cut_position
```

```
    return  q, s[n].

GET - CUTS (s, n)
    cuts = []
    while n ≥ 0:
        cut.append s[n]
        n - = s[n]
    return  cuts
```

3.

```
    let F[0:n] be a new array
    for i = 0 to n
        F[i] = 0
    F[1] = 1
    index = 1
    FIBONACCI (n)
        if  n ≤ index
                return F[n]
        for i = index+1  to  n
            F[i] = F[i-1] + F[i-2]
        return F[n].
```



$V = n+1$

$E = n + n-1 = 2n-2$

4. (a)

$$A[k] = \begin{cases} A[k-1] & \text{not contain the shares} \\ B[k-1] + a[k] & \text{contain the shares} \end{cases}$$

$$B[k] = \begin{cases} B[k-1] + a[k] & \text{not share} \\ a[k] & \text{buy the share on } k. \end{cases}$$

```
MAX-RETURN (a)
        if $n \le 1$      return 0
        let $A[0, \cdots n]$, $B[0 \cdots n]$  be new array
        $A[0] = 0$
        $B[0] = -\infty$
        let $C[0, \cdots n]$ be a new array
        for i from 1 to n
                if $B[i-1] > 0$            $B[i] = B[i-1] + a[i]$
                else $B[i] = a[i]$        ;      $C[i] = i$.
        buy_day = sell_day = 0.
        for i from 1 to n
                if $A[i-1] > B[i-1] + a[i]$        $a[i] = A[i-1]$
                else    $A[i] = B[i-1] + a[i]$
                        buy_day = $C[i-1]$.
                        sell_day = i
       return $A[n]$, buy_day, sell_day.
```

(b) $T(n) = O(1) + O(1) + \cdots + O(n) + O(n) = O(n)$