Algorithm Assignment 3

> *10185101210 陈俊潼*

9.3-5

The black box median program is denoted as `MEDIAN(A, p, r)`. The pseudocode is shown as follows.

```
LIEANER_SELECTION(A, p, r, i)
    if p == r
        return A[p]
    // Get the median number position
    mp = MEDIAN(A, p, r)
    // Partition with mp as pivot, which is a linear time solution
    q = PARTITION(A, p, r, mp)
    k = q - p + 1
    if k == i:
        return A[k]
    else if i < k:
        LINEAR_SELECTION(A, p, q - i, i)
    else
        LINEAR_SELECTION(A, q + 1, r, i -k)
```

9.3-7

Using the linear comlexity algorithm `RANDOMIZED_SELECT` to find the median p of the array first, Then create a new array A', all of which the value is $|a_n - p|$. Then find the k^{th} smallest element K, also using `RANDOMIZED_SELECT`. Finally, iterate the new array A' and select elements whose value is no greater than K.

The total complexity is $O(n) + O(n) + O(n) = O(n)$.

15.1-3

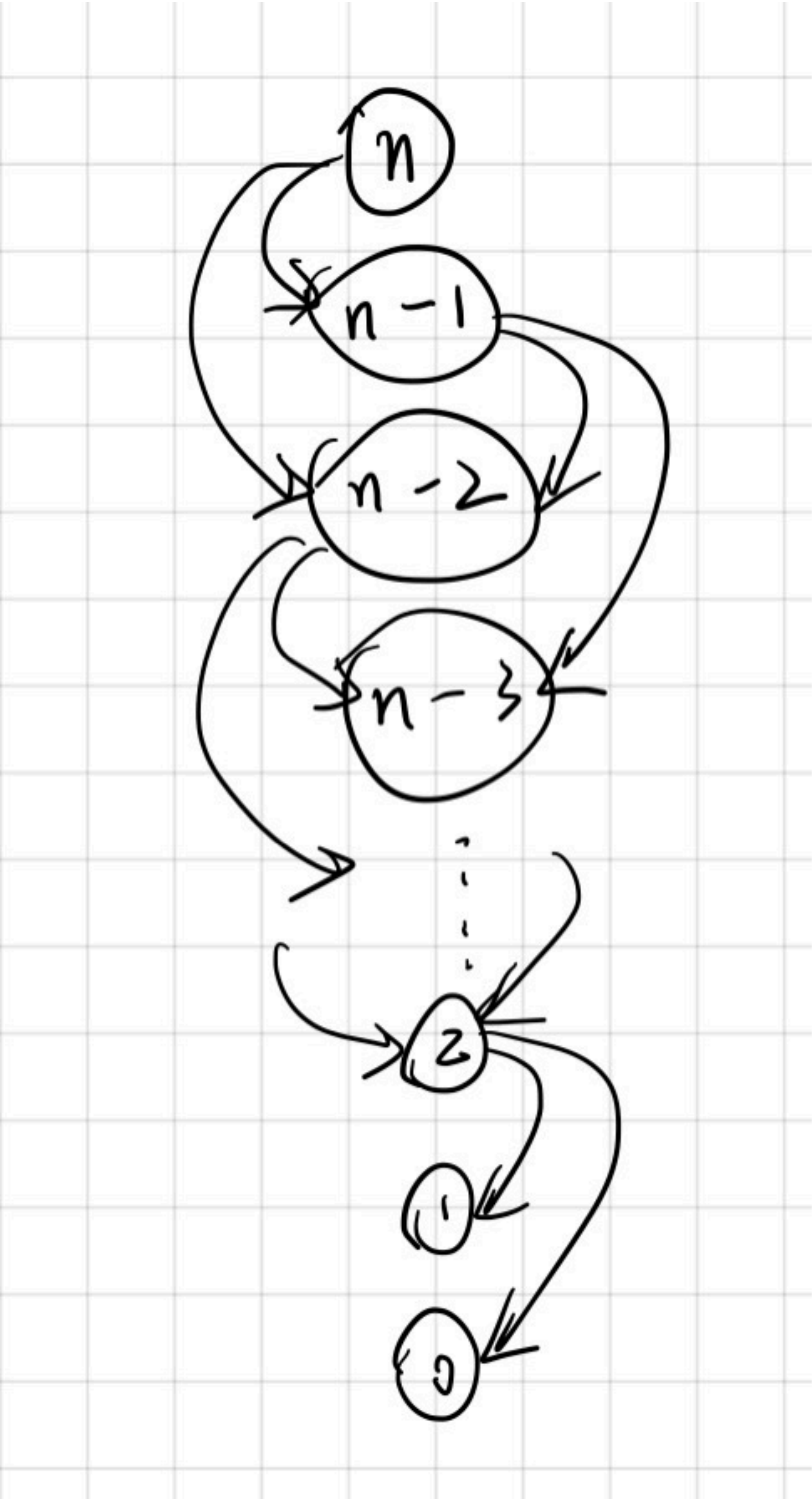
```
CUT_ROD_WITH_CUTTING_PRICE(p, n, c)
    let r[0..n] be a new array
    r[0] = 0
    for j = 1 to n
        q = -inf
        for i = 1 to j - 1
            q = max(q, p[i] + r[j-1] - c)
        r[j] = q
        else

```

15.1-5

```
FIBONACCI(n)
    let n[0..n] be an array
    n[0] = 0
    n[1] = 1
    if n <= 1
        return 1
    else
        for i = 2 to n
            n[i] = n[i - 1] + n[i - 2]
        return n[i]
```

The subproblem graph is denoted as follows:



In which there are n + 1 verteces and 2 * (n-1) edges.