Design of Algorithms by Induction for Some Problems

Dr. Gur Saran Adhar

Approach: Design by Induction-
Evaluating Polynomials

Problem: Given a sequence of real numbers $a_n, a_{n-1}, \ldots a_0$, and a real number x, compute the value of the polynomial
$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$.

Reference Udi Manber page 92

Algorithm Polynomial Evaluation$(\overline{a}, x))$
Input: $\overline{a} = a_0, a_1, \ldots, a_n$ (coefficient of polynomial,
and x (a real number)
Output: P (the value of the polynomial at x)

begin
    $P = a_n$
    for $i = 1$ to n do
        $P = x * P + a_{n-i}$
end.

Reference Udi Manber page 94

Approach: Design by Induction-
Maximum Consecutive Subsequence

Problem: Given a sequence $x_1, x_2, \ldots, x_n$ of real numbers (not necessarily positive) find a subsequence $x_i, x_{i+1}, \ldots, x_j$ (of consecutive elements) such that the sum of the numbers in it is maximum over all subsequences of consecutive elements.

Example:
In the sequence $(2, -3, 1.5, -1, 3, -2, -3, 3)$, the maximum subsequence is $(1.5, -1, 3)$; its sum is 3.5.

If all the numbers are negative, then the maximum consecutive subsequence is empty and the sum of the empty subsequence is zero.

`Reference Udi Manber page 106`

Maximum Consecutive Subsequence(X, n)
Input: x an array of size n
Output: Global_ Max
(The sum of the maximum subsequence)

```
begin
        Global_Max = 0
        Suffix_Max = 0
        for i = 1 to n do
             if x[i] + Suffix_Max > Global_Max then
                Suffix_Max = Suffix_Max + x[i]
                Global_Max = Suffix_Max
             else if x[i] + Suffix_Max > 0 then
                Suffix_Max = x[i] + Suffix_Max
                else Suffix_Max = 0
end.
```

reference Udi Manber page 107