Algorithms and Data Structures (WS15/16)

Example Solutions for Unit 7

Question 1

We will use here the following fact that was mentioned in class:
Polynomial division can be done in $O(n \log n)$.
The trick is to accelarate the version of the algorithm given in class by computing in the beginning all the multiplications that we will need.
For this cause we will present the following routine:

Compute multiplication tree

Input: n real numbers: $a_1, a_2, ..., a_n$
Output: A complete binary tree of depth $\log n$ such that each node v of the tree stores a polynomial p_v according to the following:
1. In the leaves the polynomials $x - a_i$ are stored
2. Every internal node v with children r and l satisfies $p_v = p_r \cdot p_l$
The algorithm uses a simple divide and conquer scheme:

1. If $n = 1$ then return a single node v with a polynomial $p_v = x - a_n$

2. Otherwise, solve recursively for $a_1, ..., a_{\frac{n}{2}}$ and $a_{\frac{n}{2}+1}, ..., a_n$.
Let r_1, r_2 be the roots of trees that we computed in the recursion. Create a node r with $p_r = p_{r_1} \cdot p_{r_2}$ and connect him as a father of r_1, r_2.

By A-B thm the above algorithm computes the tree in $\theta(n \log^2 n)$ time complexity.

Evaluate polynomial at n different points

Input: n real numbers: $a_1, a_2, ..., a_n$ and a polynomial $p(x)$
Output: $p(a_1), ..., p(a_n)$
Steps of the algorithm:
1. Compute the multiplication tree for $a_1, a_2, ..., a_n$.
2. Extract from the tree the polynomials $a(x) = (x-a_1)...(x-a_{\frac{n}{2}}), b(x) = (x-a_{\frac{n}{2}+1})...(x-a_n)$
3. Compute $p_{left}(x) := p \mod a, p_{right}(x) := p \mod b$
4. Compute recursively for $p_{left}, a_1, ..., a_{\frac{n}{2}}$ and for $p_{right}, a_{\frac{n}{2}+1}, ..., a_n$
Important: In the recursion we extract all the multiplications we need from the tree. This extraction costs us $O(n)$ time.
By A-B thm - steps 2,3,4 costs us $O(n \log n)$ time. As mentioned before, step 1 costs $O(n \log^2 n)$.
Thus the whole algorithm takes $O(n \log^2 n)$ time and we are done.

Question 2

Let's reduce the problem to polynomial evaluation in the following way. First consider a polynomial: $f(x) = (x+1)(x+2)...(x+\sqrt{n})$. Evaluation of $f(x)$ in $x = 0$ gives to us a product $1 \cdot 2 \cdot 3 \cdot \sqrt{n}$. Search for first integer a, such that $(a+1)^2 > n$. Clearly it can be done in $\mathrm{O}(n)$ time. If we evaluate $f(x)$ in $x = \sqrt{n}$, it gives product from $\sqrt{n}+1$ to $2\sqrt{n}$. Acting in the same manner we can see that our initial problem maps to evaluation of polynomial degree $\sqrt{n}$ in $\sqrt{n}$ points. $T(n) = \mathrm{O}(\sqrt{n}\log\sqrt{n}) = \mathrm{O}(\sqrt{n}\log n)$.