CS4102 - ALGORITHMS - S'16 — ABHI SHELAT
DUE WED FEB 24, 2016 AT 5P VIA SUBMISSION SITE

H3.AS4BD

You may collaborate with other students on the homework but you must submit your own individually written solution, identify your collaborators, and acknowledge any external sources that you consult. Do no submit a solution that you cannot explain to me. Use *exactly* 2 pages for each answer.

PROBLEM 1 *Approximate Square Root*

Present an algorithm that on input $n \in \mathbb{N}$, outputs $\lfloor \sqrt{n} \rfloor$ using $O(\log(n))$ integer ops.

PROBLEM 2 *Why 5 in Median?*

Recall the deterministic selection algorithm:

SELECT$(A[1,\ldots,n], i)$

1 Base case if $|A| < 5$.
2 $p \leftarrow$ MEDIANOFMEDIANS(A)
3 $A_\ell, A_r, i_p \leftarrow$ PARTITION(A, p)
4 **if** $i_p = i$ **return** $A[i_p]$
5 **elseif** $i_p < i$ **return** SELECT$(A_r, i - i_p)$
6 **else** **return** SELECT$(A_\ell, i_p - i)$

MEDIANOFMEDIANS$(A[1\,..\,n])$

1 Divide A into lists of 5 elements. If only one element, return it.
2 Compute the median of each small list, store these medians in a new list B
3 $p \leftarrow$ SELECT$(B, \lceil n/10 \rceil)$
4 **return** p

1. Suppose Line 1 of MEDIANOFMEDIANS divides A into lists of 3 elements each instead of 5 elements and line 3 is modified to pick the $\lceil n/6 \rceil^{\text{th}}$ element. State an upper and lower bound on the size of A_ℓ. Be as precise as you can.
2. Analyze the running time of SELECT under the 3-element version of MEDIANOFMEDIANS.

PROBLEM 3 *Compute the FFT on the values* $(4,1,3,2,2,3,1,4)$. *Illustrate the steps for the first level of recursion, you can assume the base case occurs at* $n = 4$. *You can leave your answers in terms of* $\omega_1, \omega_3, \omega_5, \omega_7$, *i.e., without multiplying those roots out.*

PROBLEM 4 *Planet Laser*

The NASA Near Earth Object Program lists potential future Earth impact events that the JPL Sentry System has detected based on currently available observations. Sentry is a highly automated collision monitoring system that continually scans the most current asteroid catalog for possibilities of future impact with Earth over the next 100 years.

This system allows us to predict that i years from now, there will be x_i tons of asteroid material that has near-Earth trajectories. In the mean time, we can build a space laser that can blast asteroids. However, each laser blast will require *exajoules* of energy, and so there will need to be a recharge period on the order of *years* between each use of the laser. The longer the recharge period, the stronger the laser blast; e.g. after j years of charging, the laser will have enough power to obliterate d_j tons of asteroid material. This problem explores the best way to use such a laser.

The input to the algorithm consists of the vectors $(x_1, \ldots, x_n)$ and $(d_1, \ldots, d_n)$ representing the incoming asteroid material in years 1 to n, and the power of the laser d_i if it charges for i years. The output consists of the optimal schedule for firing the laser which obliterates the most material.

Example Suppose $(x_1, x_2, x_3, x_4) = (1, 10, 10, 1)$ and $(d_1, d_2, d_3, d_4) = (1, 2, 4, 8)$. The best solution is to fire the laser at times $3, 4$ in order to blast 5 tons of asteroids.

(a) Construct an instance of the problem on which the following "greedy" algorithm returns the wrong answer:

BadLaser$((x_1, \ldots, x_n), (d_1, \ldots, d_n))$

Compute the smallest j such that $d_j \geq x_n$. Set $j = n$ if no such j exists.
Shoot the laser at time n.
if $n > j$ then BadLaser$((x_1, \ldots, x_{n-j}), (d_1, \ldots, d_{n-j}))$.

Intuitively, the algorithm figures out how many years (j) are needed to blast all the material in the last time slot. It shoots during that last time slot, and then accounts for the j years required to recharge for that last slot, and recursively considers the best solution for the smaller problem of size $n - j$.

(b) Given an array holding x_i and d_i, devise an algorithm that blasts the most asteroid material. Analyze the running time of your solution.