COMP3821
Extension Algorithms and Programming Techniques

Charlie Bradford

April 10, 2018

1. Assume you have two arrays A and B...
2. Assume you are given an array A...
3. Let M be an $n \times n$ matrix...
4. Assume you have an array of 2n distinct integers. Find the largest and the...
5. Assume you have an array of 2^n distinct integers...
6. You are given a $2^n \times 2^n$ board with one of its cells missing...
7. Multiply the following pairs of polynomials using at most the prescribed number of where both numbers multiplied are large (large numbers are those that depend on the coefficients and can be arbitrarily large).
   (a) $P(x) = a_0 + a_2x^2 + a_4x^4 + a_6x^6$ and $Q(x)b_0 + b_2x^2 + b_4x^4 + b_6x^6 + b_8x^8$ using at most 8 large number multiplications.
   - $P(x) * Q(x) = C(x) = a_0b_0 + (a_0b_2 + b_0a_2)x^2 + (a_0b_4 + b_0a_4 + a_2b_2)x^4 + (a_0b_6 + b_0a_6 + a_2b_4 + b_2a_4)x^6 + (a_0b_8 + a_2b_6 + b_2a_6 + a_4b_4)x^8 + (a_2b_8 + a_4b_6 + b_4a_6)x^{10} + (a_6b_6 + a_4b_8)x^{12} + a_6b_8x^{14}$
   - Let $y = x^2$
   - $C(y) = a_0b_0 + (a_0b_2 + b_0a_2)y + (a_0b_4 + b_0a_4 + a_2b_2)y^2 + (a_0b_6 + b_0a_6 + a_2b_4 + b_2a_4)y^3 + (a_0b_8 + a_2b_6 + b_2a_6 + a_4b_4)y^4 + (a_2b_8 + a_4b_6 + b_4a_6)y^5 + (a_6b_6 + a_4b_8)y^6 + a_6b_8y^7$
   - Let the coefficient of y^n in c be c_n
   - $C(y) = c_0 + c_1y + c_2y^2 + c_3y^3 + c_4y^4 + c_5y^5 + c_6y^6 + c_7y^7$
   - Then we find the values of C for the 8 smallest integers
     - $C(4) = P(4) * Q(4) = c_0 + 4c_1 + 16c_2 + 64c_3 + ...$
     - $C(3) = ...$
     - $\vdots$
     - $C(-3) = ...$
   - Then we can use simultaneous equations to solve for each of the coefficients of C

   (b) $P(x) = a_0 + a_{100}x^{100}$ and $Q(x)b_0 + b_{100}x^{100}$ with at most 3 large number multiplications.
   - Let $y = x^{100}$
   - $C(x) = P(x) * Q(x) = a_0b_0 + (a_0b_{100} + a_{100}b_0)x^{100} + a_{100}b_{100}x^{200}$
   - $C(y) = a_0b_0 + (a_0b_{100} + a_{100}b_0)y + a_{100}b_{100}y^2$
   - Find the values of C for the 3 smallest integers
     - $C(1) = P(1) * Q(1) = a_0b_0 + (a_0b_{100} + a_{100}b_0) + a_{100}b_{100}$
     - $C(0) = P(0) * Q(0) = a_0b_0$
     - $C(-1) = P(-1) * Q(-1) = a_0b_0 - (a_0b_{100} + a_{100}b_0) + a_{100}b_{100}$
   - Then solve for the coeffcients of C
     - $a_0b_0 = C(0)$
     - $(a_0b_{100} + a_{100}b_0) = \frac{C(1)-C(-1)}{2}$
     - $a_{100}b_{100} = \frac{C(1)+C(-1)}{2} - C(0)$
8. (a) Find the square of complex number $z = a + ib$ using only two real number multiplications plus as many additions and subtractions as you wish.

$$\begin{aligned} z^2 &= a^2 - b^2 + 2abi \\ &= (a+b)(a-b) + (a*b + a*b) \end{aligned}$$

(b) Multiply two complex numbers $z_1 = a + ib$ and $z_2 = c + id$ using only three real number multiplications.

$$\begin{aligned} z_1 z_2 &= (a+ib)(c+id) \\ &= ac - bd + (ad+bc)i \\ ac &= a * c \\ bd &= b * d \\ ad + bc &= (a+b) * (d+c) - ac - bd \end{aligned}$$

9. Describe all k which satisfy $i\omega_{64}^{13}\omega_{32}^{11} = w_{64}^{k}$.

10. Consider the polynomial
$$P(x) = (x - w_{64}^{0})(x - w_{64}^{1})(x - w_{64}^{2})...(x - w_{64}^{63})$$
   (a) Compute P(0); $P(0) = \omega_{64}^{32*63} = -1$
   (b) What is the degree of $P(x)$? What is its coeffecient of the highest degree of x present in $P(x)$? The degree is 64 the coeffecient of the highest degree of x is 1.
   (c)

11. Describe how you would compute all elements of the sequence $F(0), F(1), F(2), ..., F(2n)$ where
$$F(m) = \sum_{\substack{i+j=m \\ 0 \le i,j \le n}} log(j+2)^{i+1}$$
   in time $O(n\log n)$. By rearranging we have
$$F(m) = \sum_{\substack{i+j=m \\ 0 \le i,j \le n}} (i+1)log(j+2)$$

12. In Elbonia coin demoninations are 81c, 27c, 9c, 3c, 1c. Design an algorithm that, given the amount is a multiple of 1c, pays it with a minimal number of coins.

   We take a greedy approach, using the largest denomination we can at the time.

13. Give an example of a denominations containing the single cent coin for which the greed algorithm does not always produce an optimal solution.

   Take the set 12c, 10c, 1c. To make 100c the greed algorithm would use 8*12c and 4*1c for a total of 12 coins, whereas an optimal approach would use 10*10c.

14. Assume you are given n tasks each of which takes the same, unit amount of time to complete. Each task i has an integer deadline, d_i and penalty p_i associated with it which you pay if you do not complete the task in time. Design an algorithm that schedules the tasks so that the total penalty you have to pay is minimised.

   Aim to complete all tasks by their deadline, if multiple tasks clash chose the one with higher penalty and move the others to the next available spots before the deadline. If there are no spots available, complete them at the end.

15. There is a line of 111 stalls, some of which need to be covered with boards. You can use up to 11 boards, each of which may cover any number of consecutive stalls. Cover all the necessare stalls while covering as few stalls as possible.

   Use one board that covers all the stalls. Find the largest stretch of stalls that don't need to be covered and cut the board out from that area. Repeat until you have 11 boards.

16. You are running a small manufacturing show with plenty of workers but with a single milling machine. You have to produce n items; each ite i requires m_i machining time first then p_i polishing time by hand. The machine can mill only one object at a time, but your workers can polish in parallel as many objects as you wish. You have to determine the order in which the objects should be machine so that the whole production sis finished as quickly as possible.

   Order the items in order of non-increasing polishing time.

17. You are given a set S of n overlapping arcs of the unit circle. The arcs can be of different lengths. Find a largest subset P of these arcs such that no two arcs in P overlap. Prove that your solution is optimal.

    Pick an arc. Select the the arc that starts after this arc and has the closest finish. Continue doing this until you get back the the beginning. Repeat for all arcs and choose the largest S.

18. You are given a set S of n overlapping arcs of the unit circle. The arcs can be of different lengths. YOu have to stab these arcs with the minimal number of needles so that every arc is stabbed at least once. In other words, you have to find a set of aas few points on the unit circle as possible so that every arc contains at least one point. Prove that your solution is optimal.

    Pick an arc. Stab it in the end. Remove all stabbed arcs and continue until all arcs are gone. Repeat for all intervals and select the solution with the fewest needles.
    To prove this solution correct, consider an optimal solution. Move all needles to the right until they stab any of their arcs at the right most end. This is the solution that would have been optained starting with any of those arcs.

19. Let X be a set of n intervals on the real line. A subset of intervals $Y \subseteq X$ is called a tiling path if the intervals in Y cover the intervals in X. The size of a tiling cover is just the number of intervals. Design and estimate the time complexity of an alforithm to compute the smallest tiling path of X as quickly as possible. Assume that your input constist of two arrays $X_L[1..n]$ and $X_R[1..n]$, representing the left and right endpoints of the intervals in X.

    Start at the first interval in X. Pick the interval that starts before x finishes and has the farthest endpoint. Repeat for the selected interval until the subset is spanned. Complexity: $O(n^2)$

20. Suppose you have n video streams that need to be sent, one after another, over a communication link. Stream i consists of a total of b_i bits that need to be sent, at a constant rate, over a period of t_i seconds. You cannot send two streams at the same time, so you need to determine a schedule for th estreams: an order in which to send them. Whichever order you choose, there cannot be any delays between the end of one stream and the start of the next. Suppose you schedule starts at time 0. And therefore ends at time $\sum_{i=1}^{n} t_i$ seconds. We assume that all the values b_i and t_i are posiive integers. Now, because you're just one user, the linke does not want you takng up too much bandwidth, so it imposes the following constraint, using a fixed parameter r:

    For each natural number $t > 0$, the total number of bits you send over the time interval from 0 to t cannot exceed rt.

    Note that this constraint is only imposed for time intervals that start at 0, no for time intervals that start at any other value. A schedule is valid if it satisfies the constraint.

    (a) Design an $O(n\log n)$ algorithm which outputs a valid schedule if there is one and outputs the message "no valid schedule" otherwise.

        Use a sort algorithm to sort the streams in order of non-decreasing $\frac{b_i}{t_i}$. The interate through the streams and check that $\sum_{i=0}^{n-1} b_i - rt_i < 0$ if that is violated at any point then there is no valid schedule.

    (b) Design an $O(n\log n)$ algorithm.

        Consider the excess $s_i = b_i - rt_i$ of each stream. If there is a valid schedule then it can be obtained simply by placing all the streams with negative excess before all the streams with positive excess.

21. A photocopying service...

    In order to minimise $\sum_{i=0}^{n} w_i C_i$ we have to consider the time and weight of each job equaly. We can do this by sorting the jobs in order of non-decreasing $\frac{w_i}{t_i}$.

22. You are biven n points $x_i (1 \leq i \leq n)$ on the real line and n intervals $I_j = [l_j, r_j]$, $(1 \leq j \leq n)$. Design an algorithm which runs in time $O(n^2)$ and determines if each point x_i can be assigned to a distinct interval I_j so that $x_i \in I_j$.

    Sort the points in non-decreasing order. For each x_i choose all the intervals that start before it. If any of the intervals stop before x_i then it can't be mapped to a point so we return false. Otherwise we map x_i to the interval and remove it from consideration.

23. You are given a connected graph with weighted edges. Find a spanning tree such that the largest weight of all its edges is as small as possible.

Sort the edges in non-decreasing order of weight. Select every edge that does not create a cycle until you get a spanning tree.

24. IN Elbonia cities are connected with one way roads and it takes one whole day to travel between any two cities. Thus, if you need to reacha city and there is no direct road, you have spend a night in a hotel in all intermediate cities. You are given a map of Elbonia with toll charges for all roads and the prices of the cheapest hotels in each city. You have to travel from capital city C to a resort city R.

    Use djikestra' algo with directed edges and the weight of each edge being the toll + the cost of the hotel at the end node.

25. Compute the DFT of ¡2, 7, 9, 10, 3, 5, 6¿

    The corresponding polynomial is

$$Q(x)2 + 7x + 9x^2 + 10x^3 + 3x^4 + 5x^5 + 6x^6$$

Group even and odd powers

$$Q(x) = 2 + 9x^2 + 3x^4 + 6x^6 + x(7 + 10x^2 + 5x^4)$$

Substituting y for x^2

$$Q(x) = 2 + 9y + 3y^2 + 6y^3 + x(7 + 10y + 5y^2)$$

Now we have

$$Q^e(x) = 2 + 9y + 3y^2 + 6y^3$$
$$Q^o(x) = 7 + 10y + 5y^2$$

With

$$Q(x) = Q^e(x) + xQ^o(x)$$

Now we split further, and substitute z for x^2

$$Q^e(x) = 2 + 3z + y(9 + 6z)$$
$$= Q^{ee}(x) + yQ^{eo}(x)$$
$$Q^o(x) = 7 + 5z + y(10)$$
$$= Q^{oe}(x) + yQ^{oo}(x)$$

We only need the 2 roots of unity to figure out the DFT of each $Q^{**}(x)$

$$DFT(\langle 2, 3\rangle) = \langle Q^{ee}(\omega_2^0), Q^{ee}(\omega_2^1)\rangle = \langle Q^{ee}(1), Q^{ee}(-1)\rangle = \langle 5, -1\rangle$$
$$DFT(\langle 9, 6\rangle) = \langle Q^{eo}(1), Q^{eo}(-1)\rangle = \langle 15, 3\rangle$$
$$DFT(\langle 7, 5\rangle) = \langle Q^{oe}(1), Q^{oe}(-1)\rangle = \langle 12, 2\rangle$$
$$DFT(\langle 10\rangle) = \langle Q^{oe}(1)\rangle = \langle 10\rangle$$

Now we find the DFT of each Q^*

We take each $\langle a, b\rangle$ and each $\langle c, d\rangle$ and find

$$\langle a + \omega_4^0 c, b + \omega_4^1 d, a - \omega_4^0 c, b - \omega_4^1 d\rangle DFT(\langle 2, 9, 3, 6\rangle) = \langle 5 + \omega_4^0 15, -1 + \omega$$

$$DFT(\langle 2, 9, 3, 6\rangle) = \langle 5 + 1 * 15, -1 + i * 3, 5 - 1 * 15, -1 - i * 3\rangle$$
$$DFT(\langle 2, 9, 3, 6\rangle) = \langle 20, -1 + 3i, -10, -1 - 3i\rangle$$
$$DFT(\langle 7, 10, 5, 0\rangle) = \langle 12 + \omega_4^0 10, 2 + \omega_4^1 0, 12 - \omega_4^0 10, 2 - \omega_4^1(0)\rangle$$
$$DFT(\langle 7, 10, 5\rangle) = \langle 12 + 1 * 10, 2 + i * 0, 512 - 1 * 10, 2 - i * 0\rangle$$
$$DFT(\langle 7, 10, 5\rangle) = \langle 22, 2, 2, 2\rangle$$

$$DFT(\langle 2, 7, 9, 10, 3, 5, 6\rangle = \langle 20 + \omega_8^0(22), -1 - 3i + \omega_8^1(2), -10 + \omega_8^2(2), -1 - 3i + \omega_8^3(2),$$
$$20 - \omega_8^0(22), -1 - 3i - \omega_8^1(2), -10 - \omega_8^2(2), -1 - 3i - \omega_8^3(2)\rangle$$
$$=\rangle 42, -1 + 3i + (\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i)(2), -10 + 2i, -1 - 3i + (-\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i)(2),$$

Extra Questions Design an algorithm which uses one fair die to generate 5 equally likely outcomes, and analyse the expected runtime of your algorithm.

Roll the die, if the number < 5 then the outcome is the number shown, else reroll. Expected number of rolls:

$$\sum_{k=1}^{\infty} k(\frac{5}{6^k}) = 5\sum_{k=1}^{\infty} \frac{k}{6^k} = 5(\frac{\frac{1}{6}}{1-\frac{1}{6}})(\frac{1}{1-\frac{1}{6}}) = 5(\frac{1}{5})(\frac{6}{5}) = \frac{6}{5}$$