Midterm Review Problems 2017

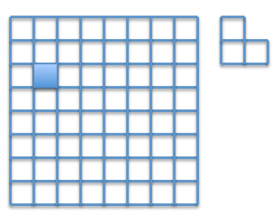
Simple stuff such as binary search etc.

1. Assume you are given two arrays A and B, each containing n distinct numbers and the equation $x^8 - x^4y^4 = y^6 + x^2y^2 + 10$. Design an algorithm which runs in time $O(n \log n)$ which finds if A contains a value for x and B contains a value for y that satisfy the equation.

2. Let M be an $n \times n$ matrix of distinct integers $M(i, j)$, $1 \leq i \leq n$, $0 \leq j \leq n$. Each row and each column of the matrix is sorted in the increasing order, so that for each row i, $1 \leq i \leq n$,
$$M(i, 1) < M(i, 2) < \ldots < M(i, n)$$
and for each column j, $1 \leq j \leq n$,
$$M(1, j) < M(2, j) < \ldots < M(n, j)$$
You need to determine whether M contains an integer x in $O(n)$ time.

Divide And Conquer

3. You are given a $2^n \times 2^n$ board with one of its cells missing (i.e., the board has a hole); the position of the missing cell can be arbitrary. You are also given a supply of "dominoes" each containing 3 such squares; see the figure:



Your task is to design an algorithm which covers the entire board with such "dominoes" except for the hole.

Polynomial Multiplication

4. Multiply the following pairs of polynomials using at most the prescribed number of multiplications of large numbers (large numbers are those which depend on the coefficients and thus can be arbitrarily large).

   (a) $P(x) = a_0 + a_2x^2 + a_4x^4 + a_6x^6$; $Q(x) = b_0 + b_2x^2 + b_4x^4 + b_6x^6$ using at most 7 multiplications of large numbers;

(b) $P(x) = a_0 + a_{100}x^{100}$ and $Q(x) = b_0 + b_{100}x^{100}$ with at most 3 multiplications of large numbers.

FFT, convolution and related concepts

5. Describe all k which satisfy $i\,\omega_{64}^{13}\omega_{32}^{11} = \omega_{64}^{k}$ (i is the imaginary unit).

6. Compute all elements of the sequence $F(0), F(1), F(2), \ldots, F(2n)$ where

$$F(m) = \sum_{\substack{i+j=m\\ 0\le i,j\le n}} \log(j+1)^i$$

in time $O(n \log n)$.

Greedy and related concepts (spanning trees etc)

7. There are N robbers who have stolen N items. You would like to distribute the items amongst the robbers (one item per robber). You know the precise value of each item. Each robber has a particular range of values they want their item to be worth (too cheap and they will not have made money, too expensive and they will draw a lot of attention). Devise an algorithm that can distribute the items so each robber is happy or determines that there is no such distribution.

8. Assume that a weighted (undirected) graph $G = (V, E)$ has all weights of edges distinct and that its set of vertices V has been partitioned into two disjoint subsets, X and $V \backslash X$ and assume that an edge $e = (u, v)$ is the smallest weight edge whose one end belongs to X and the other end to $V \backslash X$. Prove that every spanning tree must contain edge e.

9. Let $G = (V, E)$ be a weighted (undirected) graph containing a cycle C has all weights of edges distinct and let e be the highest weight edge in C. Prove that e cannot belong to the minimum spanning tree.

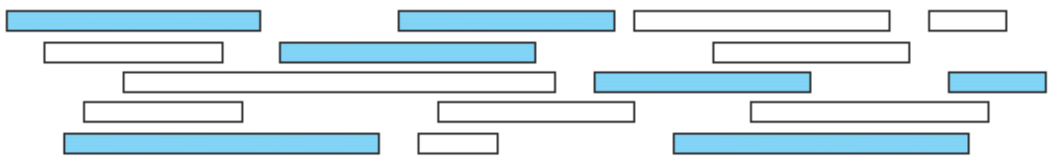
10. Assume that you are given a weighted (undirected) graph $G = (V, E)$ with all weights of edges distinct and its minimum spanning tree T. Assume now hat you add a new edge e to G. Design a linear time algorithm which produces the minimum spanning tree for the new graph with the additional edge.

11. Assume you have \$2, \$1, 50c, 20c, 10c and 5c coins to pay for your lunch. Design an algorithm that, given the amount that is a multiple of 5c, pays it with a minimal number of coins.

12. Assume denominations of your $n+1$ coins are $1, c, c^2, c^3, \ldots, c^n$ for some integer $c > 1$. Design a greedy algorithm which, given any amount, pays it with a minimal number of coins.

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

14. 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, that is, any real value that is contained in some interval in X is also contained in some interval in Y. The size of a tiling cover is just the number of intervals. Describe and analyse an algorithm to compute the smallest tiling path of X as quickly as possible. Assume that your input consists of two arrays $X_L[1..n]$ and $X_R[1..n]$, representing the left and right endpoints of the intervals in X.



A set of intervals. The seven shaded intervals form a tiling path.

15. 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 the streams: 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 your schedule starts at time 0 (and therefore ends at time $\sum_{i=1}^{n} t_i$, whichever order you choose). We assume that all the values b_i and t_i are positive integers. Now, because youre just one user, the link does not want you taking 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 $r\,t$.

Note that this constraint is only imposed for time intervals that start at 0, not for time intervals that start at any other value. We say that 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.

(b) Design an $O(n)$ such algorithm.

16. A photocopying service with a single large photocopying machine faces the following scheduling problem. Each morning they get a set of jobs from customers. They want

to do the jobs on their single machine in an order that keeps their customers happiest. Customer is job will take t_i time to complete. Given a schedule (i.e., an ordering of the jobs), let C_i denote the finishing time of job i. For example, if job j is the first to be done we would have $C_i = t_j$, and if job j is done right after job i, we would have $C_j = C_i + t_j$. Each customer i also has a given weight w_i which represents his or her importance to the business. The happiness of customer i is expected to be dependent on the finishing time of is job. So the company decides that they want to order the jobs to minimize the weighted sum of the completion times, $\sum_{i=0}^{n} w_i C_i$. Design an efficient algorithm to solve this problem. That is, you are given a set of n jobs with a processing time t_i and a weight w_i for job i. You want to order the jobs so as to minimize the weighted sum of the completion times, $\sum_{i=0}^{n} w_i C_i$.

17. You are given 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 $x_i \in I_j$.

18. You are given a collection C of intervals by their end points; design an algorithm which finds a sub-collection of C which consists of disjoint intervals and has the largest number of intervals from C.

Additional problems for extended classes only (COMP3821/9801)

19. (Warning: a tricky one!) Assume that you are given an array A containing $2n$ numbers. The only operation that you can perform is make a query if element $A[i]$ is equal to element $A[j]$, $1 \leq i, j \leq 2n$. Your task is to determine if there is a number which appears in A at least n times using an algorithm which runs in linear time.