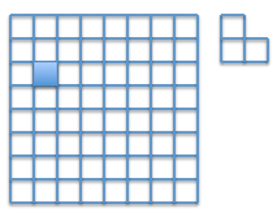
Midterm Review Problems 2016

1. Given n real numbers $x_1, ..., x_n$ where each x_i is a real number in the interval $[0, 1]$, devise an algorithm that runs in linear time and that will output a permutation of the n numbers, say $y_1, ..., y_n$, such that $\sum_{i=2}^{n} |y_i - y_{i-1}| < 1.0001$.

2. 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.

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

4. 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.

5. Design an algorithm which multiplies a polynomial of degree 16 with a polynomial of degree 8 using only 25 multiplications in which both operands (which both depend on the coefficients of the polynomial) can be arbitrarily large.

6. 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.

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

8. What are the real and the imaginary parts of $e^{i\,\frac{\pi}{4}}$? Compute the DFT of the sequence $(1, 2, 3, 4, 5, 6, 7)$ by applying the FFT algorithm by hand.

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

$$F(j) = \sum_{\max(0,j-n) \leq i \leq \min(n,j)} i^3(j-i)^2$$

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

10. You are given a connected graph with weighted edges with all weights distinct. Prove that such a graph has a unique spanning tree.

11. Assume that you are given a complete graph G with weighted edges such that all weights are distinct. We now obtain another complete weighted graph G' by replacing all weights $w(i,j)$ of edges $e(i,j)$ with new weights $w(i,j)^2$.

(a) Assume that T is the minimal spanning tree of G. Does T necessarily remain the minimal spanning tree for the new graph G'?

(b) Assume that p is a shortest path from a vertex u to a vertex v in G. Does p necessarily remain the shortest path from u to v in the new graph G'?

12. Assume that you are given a complete weighted graph G with n vertices $v_1, \ldots, v_n$ and with the weights of all edges distinct. Assume that you are also given the minimal spanning tree T for G. You are also given an additional new vertex v_{n+1} and weights $w(n+1, j)$ of all new edges $e(n+1, j)$ between the new vertex v_{n+1} and all old vertices $v_j \in G$, $1 \leq j \leq n$. Design an algorithm which produces a minimum spanning tre T' for the new graph containing the additional vertex v_{n+1} and which runs in time $O(n)$.

13. Along the long, straight road from Loololong to Goolagong there are n radio towers on a straight line. Each tower has an integer position and an integer radius of range. When a tower is activated, any towers within the radius of range of the tower will also activate, and those can cause other towers to activate and so on. We need an algorithm which finds the fewest number of towers you must activate to cause all towers to activate. Someone has proposed the following two algorithms:

- Algorithm 1: Find the unactivated tower with the largest radius (if multiple with the same radius, pick the one with smallest position). Activate this tower. Find and remove all activated towers. Repeat.
- Algorithm 2: Find any unactivated tower which is outside of the radius of all the other unactivated towers and activate it. If there is none, activate the leftmost tower. Repeat.

(a) Give examples which show that neither Algorithm 1 nor Algorithm 2 solve the problem correctly.

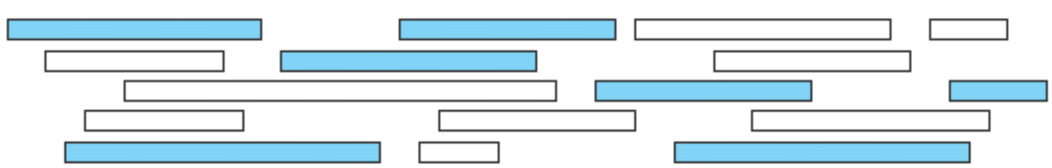
(b) Design an algorithm which correctly solves the problem.

14. 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.

15. 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.

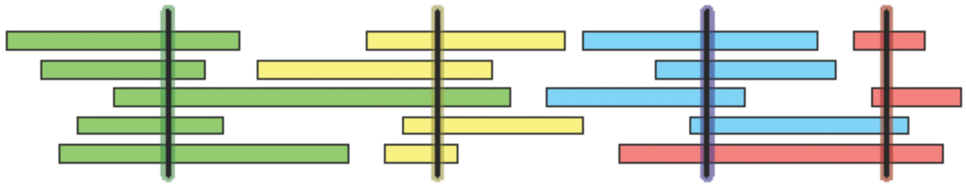
16. Give an example of a set of denominations containing the single cent coin for which the greedy algorithm does not always produce an optimal solution.

17. 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.

18. Let X be a set of n intervals on the real line. We say that a set P of points stabs X if every interval in X contains at least one point in P; see the figure below. Describe and analyse an efficient algorithm to compute the smallest set of points that stabs X. 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 stabbed by four points (shown here as vertical segments)

19. Assume you are given n sorted arrays of different sizes. You are allowed to merge any two arrays into a single new sorted array and proceed in this manner until only one array is left. Design an algorithm that achieves this task and uses minimal total number of moves of elements of the arrays. Give an informal justification why your algorithm is optimal.

Additional problems for extended classes only (COMP3821/9801)

20. 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.

21. Assume that you are given a complete binary tree with $2^n - 1$ many nodes, each node containing a distinct number. A node is a *local minimum* if the number it contains is smaller than the numbers contained in any of the nodes connected to it via an edge. Thus, the root would be a local minimum if its number is smaller than the numbers contained in both of its children; a leaf would be a local minimum if the number it contains is smaller than the number contained in its parent; any other number is a local minimum if the number it contains is smaller that numbers contained in its parent and both of its children. You can make queries by specifying a node; the query returns the value stored at that node. Design an algorithm which finds a local minimum (not all, just one) and makes only $O(n)$ queries.