Summary for midterm

To study for the midterm, you should go through your lecture notes and the problem sets we did during the sections. The CLRS book contains very good exercises for practice at the end of every section. Further you can look at the exercises here. In general, make sure you know how the algorithms work, how to apply them to a given instance of the problem and how to compute their running times.

The chapters that were covered are: CLRS 2, 3, 4.2-4.5, 6, 7, 9, 11, 15.2, 15.4 17.1, 17.2, 18, 21, 22.

Be aware that this list may be incomplete, cross check with your lecture notes.

You can use any hand written material during the midterm.

1. Put the following expressions in increasing order of magnitude $5n^{0.01} + 100$, $2^n + 1$, $\log n^2$, $2^{n+1} - 200$, $(n + 1) \log n$, $n^{0.02} - 10$, $0.1n^{0.01}$, $\log n$.

2. Give a closed formula for the following recursions. In some cases you can use the Master theorem, in other cases you need to use the substitution method or the recursion tree method to make the appropriate guess.

\[
T(n) \leq T(n-1) + n
\]
\[
T(n) \leq n + 2T(n/2)
\]
\[
T(n) \leq T(n-1) + \log n
\]
\[
T(n) \leq 2T(n-1) + n
\]
\[
T(n) \leq 3T(n/2)
\]

3. What are the worst case running times for merge sort, linear sort, binary sort and quick sort on $n$ elements? Write the recursions corresponding to these algorithms.

4. (CLRS 6.2-6) Show that the worst-case running time of Max-Heapify on a heap of size $n$ is $\Omega(lg n)$. Give an example where the worst-case scenario happens.

5. Build a heap containing values $<3, 27, 34, 2, 1, 3, 3, 18, 5, 6>$. Delete the 4th largest element. Delete key 6.

6. Describe how to implement a priority-queue with help of a max-heap.

7. Show how to implement a first-in first-out queue and a stack with help of a priority queue.

8. Use the randomized version of quick sort to sort the elements in the following array $<10, 9, 8, 7, 6, 5, 4, 3, 2, 1>$. Show every random decision. What was the running time of this particular run? Compare this to the worst-case running time of the deterministic version of quick sort.
9. (CLRS 7.2-2) What is the running time of quick sort if all elements of an array $A$ have the same value?

10. Use Randomized-Select to find the 5th smallest value in the array $<17,3,8,7,4,9,18>$. 

11. Use the fast multiplication method to compute $9876 \times 5432$. 

12. Insert the keys $<53,11,9,72,3,17,19,29,111>$ into a hash-table using $h(k) = k \mod 5$ as the hash function. Show an example of an array of 6 numbers that result in the best and worst-case chain length with this hash function.

13. Let $h_b(<a_0 a_1 \ldots a_{n-1}>)$ be the universal hash function described in problem CLRS 11.3-6. Thus

$$h_b(<a_0 a_1 \ldots a_{n-1}>) = \sum_{i=1}^{n-1} a_i b^i \mod p$$

where $p = 5$ and $b = 4$. Show the result of inserting the following keys into a hash table of length 5. $<17893>, <87139>, <17894>, <12345>, <27893>$. 

14. What is the best order to multiply a sequence of matrices of sizes $<3,4,10,2,2,10,3>$?

15. Draw the subproblem graph corresponding to computing the longest common subsequence in $X = <1,1,0,1,1,0,1>$ and $Y = <1,1,1,0,1,1>$. How can you read the amount of computations needed to find the LCS of $X$ and $Y$ from the graph?

16. Find the longest simple path in a directed acyclic graph. (Hint: use the topological order of the nodes and define a dynamic programming algorithm.)

17. One of the most famous problems that is solved with help of a dynamic programming algorithm is the 0-1 Knapsack problem [http://en.wikipedia.org/wiki/Knapsack_problem](http://en.wikipedia.org/wiki/Knapsack_problem). In this problem we have a knapsack of a given capacity $c$ and we have $n$ items each with a value and a weight assigned. The task is to pick a subset of items that has maximum total value but the total weight does not exceed the capacity of the knapsack. This problem can be solved by creating a table $T$ of size $n \times c$, where $T[i,j]$ denotes the maximum value that can be chosen from the first $i$ items with total weight at most $j$.

Write the decision function that is used to compute every $T[i,j]$ in the table. Solve the knapsack problem for 6 items where the items have values 1,2,1,2,3,2 and weights 2,3,1,2,2,1 respectively. The maximum capacity of the knapsack is 5.

18. (CLRS 16.3-3) What is an optimal Huffman code for the following set of frequencies based on the first eight Fibonacci numbers? $a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21$.

19. (CLRS 16.3-7) Generalize Huffman’s algorithm to ternary codewords (i.e., codewords using symbols 0,1,2). Compute your code for the frequencies $f : 5, e : 9, c : 12, b : 13, d : 16, a : 45$. 


20. Let $S$ be a set (universe) of $n$ elements. Let $A_1, A_2, \ldots A_r$ be sets, such that $A_r = S$, $A_i \subseteq S$ and for every $i < j$ either $A_i \subset A_j$ OR $A_i \cap A_j = \emptyset$. Let $A = \{A_1, A_2, \ldots A_r\}$. Does the pair $(S, A)$ define a matroid? Is it a matroid if $A_r \subset S$?

21. (CLRS 18.2-4) Suppose that we insert the keys 1, 2, \ldots n into an empty B-tree of min. degree 2. How many nodes does the final B-tree have?

22. insert the numbers $< 9, 8, 5, 6, 7, 1, 3, 10, 4, 2 >$ in this order into a B-tree of minimum degree 2. How is this tree different from the tree that you get if the numbers arrive in decreasing order? Delete 10, 9, 5, 3, 1 (in this order) from the original B-tree.

23. Use the disjoint set data structure along with the union by rank and path contraction heuristics to decide whether the following graph is connected.

\[
\begin{align*}
A : B \rightarrow D & \quad E : B \rightarrow D & \quad I : H \rightarrow J \rightarrow K \\
B : A \rightarrow C \rightarrow D \rightarrow E & \quad F : G \rightarrow H & \quad J : I \rightarrow K \\
C : B \rightarrow D & \quad G : F \rightarrow H & \quad K : I \rightarrow J \\
D : A \rightarrow B \rightarrow C \rightarrow E & \quad H : \rightarrow F \rightarrow G \rightarrow I
\end{align*}
\]

You can assume that edges arrive in alphabetical order. Thus edge (A,B) is first, then (A,D) and edge (H,I) is last. How many union operations did you execute during the algorithm? What does this say about the number of connected components?

24. (CLRS 22.1-1) Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of every vertex? How long does it take to compute in-degrees?

25. (CLRS 22.1-3) Describe efficient algorithms for computing $G^T$ from $G$, for both the adjacency-list and adjacency-matrix representations of $G$. Analyze the running times of your algorithms.

26. (CLRS 22.1-5) The square of a directed graph $G(V, E)$ is the graph $G^2(V, E^2)$ such that $(u, v) \in E^2$ if and only if $G$ contains a path with at most two edges between $u$ and $v$. Describe efficient algorithms to compute $G^2$ from $G$, for both the adjacency-list and adjacency-matrix representations of $G$. Analyze the running times of your algorithms.

27. (CLRS 22.2-4) What is the running time of BFS if we represent the input graph by an adjacency-matrix and modify the algorithm to handle this form of input?

28. Find the strongly connected components of the following graph using DFS.

\[
\begin{align*}
A : B \rightarrow D & \quad E : A \rightarrow F & \quad I : H \rightarrow J \rightarrow K \\
B : A \rightarrow C \rightarrow D \rightarrow E & \quad F : G \rightarrow H & \quad J : I \\
C : D & \quad G : F \rightarrow H & \quad K : I \\
D : A \rightarrow E & \quad H : \rightarrow F \rightarrow G \rightarrow I
\end{align*}
\]

29. (CLRS 22.3-7) Rewrite the procedure DFS using a stack to eliminate recursion.