(CS1.301) Algorithm Analysis and Design (Monsoon 2022) Date: 22.09.2022

Mid-semester examination

Alloted time: 90 minutes Total marks: 25

Instructions:

- There are a total of 5 questions over 2 pages with varying credit.
- No clarifications shall be provided during the examination. In case of doubts, please make a reasonable assumption and that could be considered for grading.
- Discussions amongst the students are not allowed. No electronic devices nor notes/books of any kind are allowed.
- Any dishonesty shall be penalized heavily.
- Be clear in your arguments. Vague arguments shall not be given any credit.

Question 1

[3 marks]

Let G be a simple graph on n nodes, where n is an even number. If every node of G has degree at least n/2, then G is connected.

Question 2

[1+2+1 marks]

+2

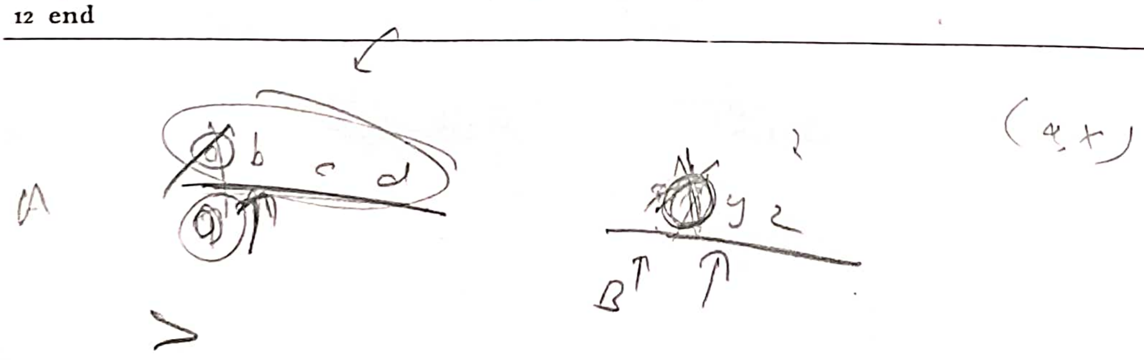
1. Compute third primitive root of unity.
2. Construct the DFT and inverse DFT matrices of order 3 × 3.
3. Compute the DFT of the vector (1, 1, 1) (without simplifying the expressions).

Question 3

[4 marks]

Algorithm 1: OneDimensionalPeak(Array A[1, n])

```
if length(A) $\leqslant$ 3 then
    return a peak here by brute-force.
end
middle ← $\left\lceil \frac{\text{length}(A)}{2} \right\rceil$;
if A[middle − 1] $\leqslant$ A[middle] and A[middle] $\overset{\geqslant}{\leqslant}$ A[middle + 1] then
    return middle;
else
    if A[middle − 1] $\geqslant$ A[middle] then
        return OneDimensionalPeak(Array A[1, middle − 1]);
    end
    return OneDimensionalPeak(Array A[middle + 1, n]);
end
```



Given an array A with n elements, we would like to find an index i of a peak element $A[i]$ where $A[i] \geqslant A[i-1]$ and $A[i] \geqslant A[i+1]$. For elements on the boundaries of the array, the element only needs to be greater than or equal to its lone neighbor to be considered a peak.

Argue the correctness of Algorithm 1 for finding a peak, and compute its running time. In particular, argue that every given array will contain a peak. Here the notation $A[i, j]$ corresponds to the sublist of elements starting from index i until the index j.

Question 4 [6 marks]

Suppose you are a winner of a Golden ticket and you are invited to a huge chocolate factory. There you are shown n different chocolates where the i'th item is worth v_i rupees, and a total quantity of w_i kilograms is available for taking. You are then handed a bag with a capacity of W kilograms, and are asked to pick any item as long as your bag was not full to its capacity. What would you do if you want to maximize the value of contents in your bag. [Assume that any fractional quantity of each chocolate can be taken, and w_i's are much smaller than W.]

Question 5 [8 marks]

Given a sequence $a_1, \ldots, a_n$, we say that two indices $i < j$ form an inversion if $a_i > a_j$, that is, if the two elements a_i and a_j are "out of order". We seek to determine the number of inversions in the sequence $a_1, \ldots, a_n$.

The basic idea is to divide the list into the two pieces $a_1, \ldots, a_m$ and $a_{m+1}, \ldots, a_n$. We first count the number of inversions in each of these two halves separately. Then we count the number of inversions (a_i, a_j), where the two numbers belong to different halves. We would like to do this merging part in $O(n)$ time.

Please read through the following code and fill in the missing snippets (in Algorithm 3) so that merging can be done in $O(n)$ time. Also construct the recursive relation for time complexity and state the correct running complexity.

Algorithm 2: Sort-and-Count(L)

```
if the list L has one element then
    return "There are no inversions";
A ← first ⌈n/2⌉ elements of L;
B ← the remaining ⌊n/2⌋ elements of L;
(r_A, A) = Sort-and-Count(A);
(r_B, B) = Sort-and-Count(B);
(r, L) = Merge-and-Count(A, B);
return r = r_A + r_B + r, and the sorted list L;
```

Algorithm 3: Merge-and-Count(A,B)

```
Maintain a Current pointer into each list, initialized to point to the front elements;
Maintain a variable Count for the number of inversions, initialized to 0;
while both the lists A and B are nonempty do
    TODO: FILL THE MISSING CODE
return Count and the merged list;
```

9 < 1

a b c d

x y 2

(CS1.301) Algorithm Analysis and Design (Monsoon 2022) Date: 19.10.2022

Deep Quiz 2

Alloted time: 45 minutes Total marks: 15

Instructions:

- There are a total of 3 questions.
- Discussions amongst the students are not allowed. No electronic devices nor notes/books of any kind are allowed.
- Any dishonesty shall be penalized heavily.
- Place your identity cards on the table for verification.
- Be clear in your arguments. Partial marking is available but vague arguments shall not be given any credit.

Kinda ✓

Question 1 [3 marks]

Let $G = (V, E)$ be a directed graph and for every edge $u \to v \in E$, a capacity $c(u \to v)$ is defined. Let $f(u \to v)$ (for all $u \to v \in E$) be a given feasible flow. Compute the residual capacities and the residual graph, with respect to f.

✓

Question 2 [5 marks]

Let n be an integer which is much larger than 5. Formulate a recursive function to compute the number of ways of providing change for n rupees only using (sufficient supply of) 1 rupee and 2 rupee coins.

Question 3 [7 marks]

Suppose you are given three strings, S_1, S_2, and S_3, where $|S_1| = n$, $|S_2| = m$, and $|S_3| = m+n$. We say that S_3 is an interleaf of S_1 and S_2 if and only if S_3 can be formed by interleaving sequences of characters from S_1 and S_2 in a way that maintains the left-to-right ordering of S_1 and S_2. For example, "split" is an interleaving of "spit" and "l", but "splti" is not, and "cchocohilaptes" is an interleaf of "chocolate" and "chips".

choco

Give an efficient dynamic programing algorithm[1] that takes S_1, S_2, and S_3 as parameters and determines whether S_3 is an in interleaf of S_1 and S_2.

Remember[redacted] - read the hints.

[1] Hint: Memoization matrix could take True or False values in each entry where True in entry $M_{i,j}$ could represent if the first $i+j$ letters of S_3 are formed by interleaving of first i letters of S_1 and first j letters of S_2.

45

(CS1.301) Algorithm Analysis and Design (Monsoon 2022)

End of Semester Examination

Alloted time: 180 minutes

Total marks: 90

Please read the instructions VERY carefully.

- There are a total of 12 questions with varying credit, printed on pages 2 till 5.
- Discussions amongst the students are not allowed. No electronic devices (including smart watches) nor notes/books of any kind are allowed.
- Any dishonesty shall be penalized heavily.
- Any theorem/lemma/claim/fact that was proved in the class can be used without proof in the exam, only by explicitly writing its statement, and a clear remark that it was chosen from the class notes.
- Questions have been framed to be disambiguous, and queries may not be answered during the examination. In case you find any ambuguity, please mention that in your answer scripts and work with it. Answers got by misreading of questions may not be given credit.
- Be clear in your arguments. Partial marking is available for every question but vague arguments shall not be given any credit.
- You do not have to write a pseudocode unless explicitly asked.
- Analysis of running times, and proofs of correctness need to be done unless explicitly asked not to.
- Questions start on the next page.
- There are hints for a couple of problems on page 5.

Question 1

[3+3 marks]

1 (i) For a very large $n \geqslant n_0$, write the following asymptotic terms in an increasing order of Big-O notation.

(a) $n^{\log(\log n)}$

(b) $n^{(\log n)^2}$

(c) 2^n

(d) $n^{\log n}$

(e) $2\sqrt{n}$ 3

(f) n^{100}

2 (ii) For n that is large enough, and constants a, b, c, compute the asymptotic growth of the following function.

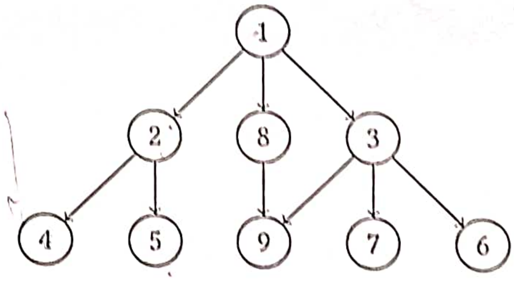
$$T(n) = a \cdot T\left(\frac{n}{b}\right) + n^c.$$

That is, what is the value of N in terms of n, a, b, c such that $T(n)$ can be written as $O(N)$.

Question 2

[3+3 marks]

Let $G = (V, E)$ be a graph as drawn below. Suppose we were to run graph search algorithms from



the node 1 to discover the node 7, what is the sequence of nodes discovered before discovering the node 7 if the search algorithm used is

(a) Breadth-first search.

(b) Depth-first search.

Assume that each of these search algorithms prioritises a neighbour of least value.

notsure b/s 1

Question 3

[8 marks]

A vertex u in a connected (undirected) graph $G = (V, E)$ is called an articulation point if removal of u from the vertex set, and removal of edges incident on the vertex u disconnects the graph. Give an algorithm to find articulation point(s) in a given graph.

Redo

Question 4 [4+2 marks]

Suppose the symbols a, b, c, d, e occur with frequencies $0.5, 0.25, 0.125, 0.0625, 0.0625$, respectively.

(a) What is the Huffman encoding of this alphabet?

(b) If this encoding is applied to a file consisting of 1,000,000 characters from a, b, c, d, e, with the given frequencies, what is the length of the encoded file in bits.

Dijkstra

Question 5 [10 marks]

A path is monotonically increasing if the weight of every edge (considered in sequence) on that path is strictly increasing. Given an edge-weighted directed graph, find a monotonically increasing shortest path from s to every other vertex.

min 7 ≤ n/2 [1 n/2 1 7 8 9

Question 6 [8 marks]

Given a sorted array of distinct integers $A[1, n]$, we want to find out whether there is an index i for which $A[i] = i$. Give a divide-and-conquer algorithm that runs in time $O(\log n)$.

Question 7 [8 marks]

A k-way merge operation is as follows. Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements (for example, in each iteration of merge sort, we do a 2-way merge operation). Give an efficient solution to this problem, using divide-and-conquer.

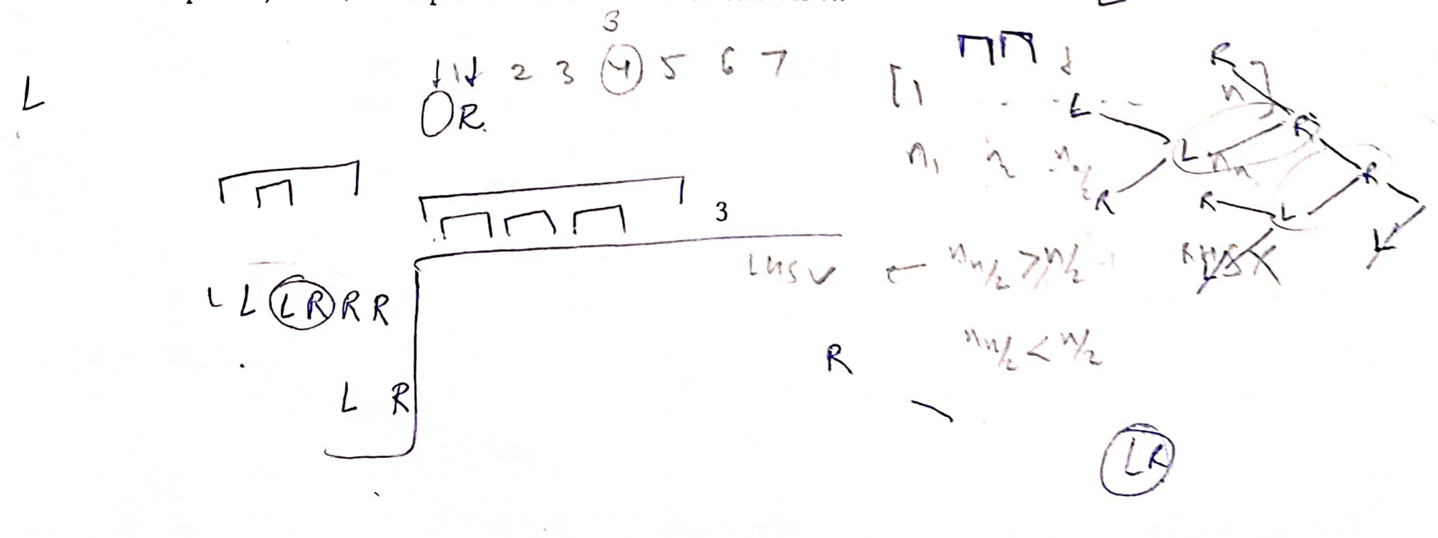
2n. 2n-2

Question 8 [6 marks]

We say that a $2n$-length string over the letters L and R is balanced if it is properly nested. L could be thought of as an opening bracket and R could be thought of as a closing bracket.
For example,

- for $n = 1$, LR (which corresponds to [ ]) is balanced but RL (which corresponds to] [ ) is not balanced.
- For $n = 2$, LRLR (which corresponds to [ ] [ ]) and LLRR (which corresponds to [[ ]]) are balanced but RLRL (which corresponds to] [ ] [ ) and RLLR (which corresponds to] [[ ]) are not balanced.

Write a recursive function (either mathematically or using pseudocode) to enumerate (or print out) all the strings of length $2n$ over the letters L, R, that are balanced. In other words, using recursion print out all the balanced $2n$-length strings over letters L, R. (No proof of correctness is required.) Here, the input to the recursive function is n.



Question 9 [10 marks]

A polygon is convex if every line that does not contain any edge of the polygon intersects the polygon in at most two points.

You are given a convex polygon P on n fixed vertices in the plane (specified by their x and y coordinates). A triangulation of P is a collection of n − 3 diagonals of P such that no two diagonals intersect (except possibly at their endpoints). Notice that a triangulation splits the polygon's interior into n − 2 disjoint triangles (see Figure 1 for a visualization).

The cost of a triangulation is the sum of the perimeters of the triangles in it. Give an efficient algorithm for finding a triangulation of minimum cost.

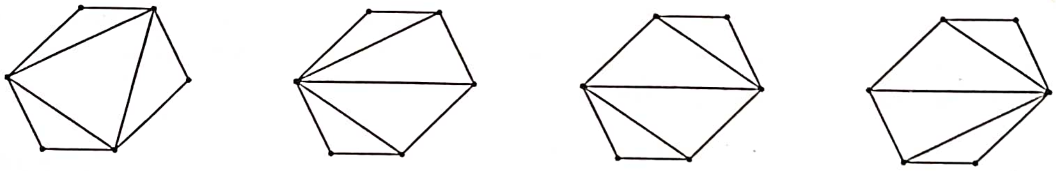


Figure 1: Some of the many different ways to triangulate a convex polygon on 6 vertices. Note that the second and the last figures are distinct.

~~Question~~ 10 [8 marks]

For the following network, with edge capacities as shown in Figure 2, find the maximum flow from s to t along with the associated minimum cut.

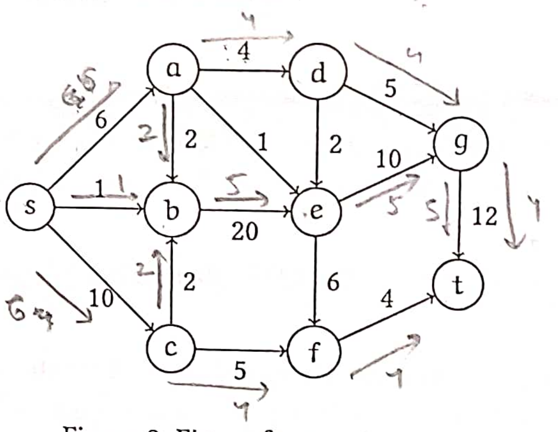


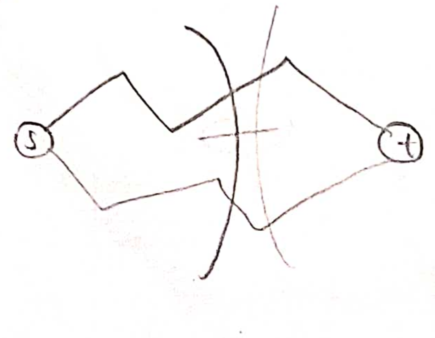
Figure 2: Figure for question 10

Please work this problem out by first starting with a feasible flow and then listing out all the intermediate steps of computing the residual graphs and augmentations until the maximum flow is found.

Question 11 [8 marks]

An edge of a flow network is called critical if decreasing the capacity of this edge (by any positive integer-valued quantity) results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

find any min cut



maxflow: mincut

so this edge is a part of every mincut

run ford fulkerson on a

find the mincut

for all edges e,

Claim this edge is a part of every mincut.

Question 12 [3+3 marks]

Argue why the following problems are in NP.

?

1. **Minimum Spanning Tree:** Given a graph $G = (V, E)$ along with edge weights $\{w_e \mid e \in E\}$, find the minimum spanning tree of G.
2. **Integer factoring:** Given integer A and k, check if there is an integer b such that $1 < b \leqslant k$, and b divides A.

Hints

- Hint for question 3: One of the search algorithms discussed in the class could be of use.
- Hint for question 9: Label the vertices of P by $\{1, 2, \ldots, n\}$, starting from an arbitrary vertex and walking clockwise. For $1 \leqslant i < j \leqslant n$, let the subproblem $A(i, j)$ denote the minimum cost triangulation of the polygon spanned by vertices $i, i + 1, \ldots, j$.