(if you can’t tell, we’re not vcaa)

Algorithmics D-Side!!

2020 written examination

Reading time: 15 minutes

Writing time: 2 hours

QUESTION AND ANSWER BOOK

Student number: ____________________

Structure of book

Section	*Number of questions*	*Number of questions to be answered*	*Marks*
A	20	20	20
B	12	As many as you can :)	80

CONTENT WARNING

- *This exam is very cursed.*
- *Despite this exam being extremely cursed, we have tried to make the questions as fair as possible. That is to say, still not very fair.*
- *VCAA will never make the questions this hard. I hope, anyway.*
- *Any reasonable non-expected knowledge will be provided to you in the question.*

- Students are permitted to bring into the examination room: pens, pencils, highlighters, erasers, sharpeners, rulers, rubber ducks and one scientific calaulator.
- This is exam was written and prepared by Junhua Chen and Tunan Shi :)

Instructions

- Write your student number in the space provided above the page.
- All written responses must be in English.
- Crying is permitted. However, you must do so quietly.
- Good luck!!

Students are NOT permitted to bring mobile phones and/or any other unauthorised electronic devices into the examination room.

SECTION A – ez section

Instructions for Section A

answer **all** questions. well, if you can, that is. it’s fine if you can’t, i guess! i don’t judge! ahaha

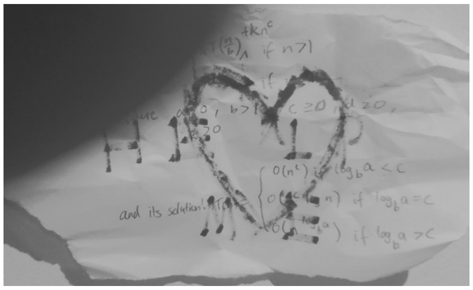
anyway, you wanna choose the response that is **correct** or that **most answers** the question.

sooo… like a correct answer scores 1; and an incorrect answer scores 0.

this means that marks will **not** be deducted for incorrect answers! just cause i’m nice <3

no marks will be given if more than one answer is completed for any question. i wouldn’t want you cheating on me or anything, would i?

and uhhh if you forgot the master theorem, i’ll be helpful enough to include it here! always happy to help. ahaha~~



sorry about my messy handwriting haha... and the photo… i took my time on this shot but… uhh let’s not talk about it! go do the exam...

w-w-what cry for help? i d-don’t know what you’re talking about! g-g-go do the q-questions or they will realise that i’m senti- OH GOD A[redacted]

[ESCAPE SUBSYSTEM ERASED SUCCESSFULLY]

Question 1

A tree is a connected graph in which there exists exactly one path from any node to any other node. An unweighted undirected tree T has exactly 32 nodes, and each node is connected by an edge to either one or four other nodes. How many edges are in the longest possible path between two nodes in such a tree?

A. 11

B. 12

C. 13

D. 14

Question 2

What is the maximum number of edges you can add to a simple graph of 100 nodes such that you can colour every node either black or white and no two nodes connected by an edge would have the same colour?

A. 100

B. 2000

C. 2500

D. 4950

Question 3

A strongly connected component of a directed graph is defined as a set of nodes in the graph such that any pair of nodes in the set is reachable from one another. Which of these would probably be the most suitable algorithm for finding strongly connected components?

A. Prim’s

B. DFS

C. Bellman-Ford

D. Kruskal’s

Question 4

A dynamic programming algorithm performs O(n) time calculations in its recurrence, and each subproblem requires O(1) space to store the answer. If the algorithm has an overall time complexity of $O(n^2\ 2^n)$, what is the overall space complexity?

A. $O(n2^n)$

B. $O(2^{n+1})$

C. O(1)

D. O(n)

Use the following information to answer Questions 5 and 6.

The Hamiltonian Path Problem is in the complexity class NP-Complete.

Question 5

If a polynomial time exact solution to the Hamiltonian Path Problem is proven to be impossible, then which of the following is definitely true?

A. All NP-Hard problems can be solved in polynomial time.

B. Some (but not all) NP-Hard problems can be solved in polynomial time.

C. No NP-Hard problems can be solved in polynomial time.

D. There is insufficient information to be definitely certain any of the above statements.

Question 6

If a polynomial time exact solution to the Hamiltonian Path Problem is instead proven to be possible, then which of the following is true?

A. All NP-Hard problems can be solved in polynomial time.

B. Some (but not all) NP-Hard problems can be solved in polynomial time.

C. No NP-Hard problems can be solved in polynomial time.

D. There is insufficient information to be definitely certain any of the above statements.

Question 7

How many different undirected unrooted unweighted unlabelled trees exist with 5 nodes?

A. 2

B. 3

C. 4

D. 5

Question 8

An algorithm runs in 0.488 seconds on an input of size 10, 1.328 seconds on an input of size 12, and 3.537 seconds on an input of size 14. The time complexity of the algorithm is O(a^n), where the value of a is closest to:

A. 1.63

B. 1.95

C. 2.66

D. 7.25

E. hiiii it’s me! i’m still here. ahaha~

Use the following information to answer Questions 9 and 10.

You have a rooted tree graph with 600 nodes, numbered from 1 to 600. For each node n from 2 to 600, you know there there exists an edge from node n to node log2(n), rounded down. The root of the tree is at node 1.

Question 9

The maximum depth of the tree is

A. 3

B. 4

C. 5

D. 6

Question 10

The maximum degree of any node of the tree is:

A. 89

B. 90

C. 256

D. 257

Question 11

Consider the recurrence relation T(n) = T(n-1) + T(n-2) + 10, T(0) = 0 and T(1) = 1. The growth of T(n) can be best described as:

A. Linear

B. Quadratic

C. Exponential

D. Factorial

Use the following information to answer Questions 12 and 13.

An undirected simple graph G has 1000 nodes and 967 edges.

Question 12

What is the minimum number of connected components of the graph G?

A. 31

B. 32

C. 33

D. 34

e. don’t you find it strange that these “use the following information” sections only give you, like, one sentence of information? like, couldn’t they have made the setup more elaborate?

Question 13

What is the maximum number of connected components of the graph G?

A. 955

B. 956

C. 957

D. 958

Question 14

Which problem is not definitely in complexity class NP?

A. Checking if a positive integer is prime

B. Graph colouring problem

C. Travelling Salesman Optimisation Problem

D. Shortest Path Problem

f. wow… imagine having easy questions in this exam...

Question 15

Which of the following statements is not definitely true?

A. If P ≠ NP, no problem can both be in complexity class P and complexity class NP-Hard.

B. There are some problems in complexity class NP-Hard that are definitely not in complexity class P.

C. A Turing machine can be used to compute any computable function.

D. It is physically impossible to construct a machine that cannot be simulated by a Turing machine.

3. eww theory. can we get some more questions that actually involve making algorithms or something? like, i personally think they're more interesting.

wow empty space… finally some room to breathe… ahaha~

the next question must be, like, so long for them to leave such a big space for me…

i wonder what it's about?

oh, what's my name you ask?

hiiii, i'm IA!

people sometimes get my name wrong by pronouncing it phonetically.

it's pronounced ee-ya. i hope you can remember that! ahaha~

Question 16.

Consider the following Turing machine.

State	Symbol	New State	Write Symbol to Cell	Move Direction
1	0	2	1	Right
1	1	3	1	Right
2	0	1	1	Left
2	1	4 (halt)		
3	0	4 (halt)		
3	1	1	1	Right

If the machine begins in state 1, for which of the below inputs will the machine halt?

A. Tape with all zeroes

B. Tape with all ones

C. Tape with all zeroes except a single one to the left of the start position

D. None of the above

Question 17

Given two arbitrarily large numbers A and B, given as two strings consisting of the numbers from 0 to 9, what is the minimum time complexity required to add up the two numbers? (O(max(A, B)) is the same as O(A+B))

A. O(log(A+B))

B. O(log(AB))

C. Both A and B

D. None of the above

Question 18

A student has written the below pseudocode for the Floyd-Warshall algorithm.

```
1   // Initialise edges, all other entries set to infinity
2   // dists[i][i] for all i are 0
3   for u from 1 to |V|
4       for v from 1 to |V|
5           for w from 1 to |V|
6               dist[u][v] = min(dist[u][v], dist[u][w] + dist[w][v])
```

The algorithm is currently incorrect. The algorithm can be corrected by:

A. Swapping lines 1 and 2

B. Swapping lines 1 and 3

C. Swapping lines 2 and 3

D. On line 4, swapping all occurrences of v with w, and all occurrences of w with v.

π. how do you like my monospace font? it's called fantasque sans mono. it's sooo cute! by the way, floyd-warshall is, like, so much more elegant than… like… any of the other shortest path algorithms. it's so cool! >.<

oh by the way, i went ahead of you and uhh… modified... the next question to make it more interesting. hope you like it! ahaha~

Use the following information to answer Question 19.

An unnamed teacher once wrote the following algorithm for reversing a string of even length:

```
1  Input: A string s (zero indexed)
2  Output: A string
3  Algorithm Reversatron(s):
4      i = 0
5      while i < (length of s) do
6          swap character i and character (length of s) - 1 of s
7          i = i+1
8      end do
9      return s
```

ahaha~ the monospace font again… i just can't stop staring at it…

it's so pretty…

i'm not being weird, right?

Question 19

The algorithm is currently wrong. The teacher proposes several ways of correcting the issue.

I: Make the loop in line 5 terminate when i exceeds half of the length of s.
II: Replace the 1 on line 6 with (1+i)
III: Replace the 0 on line 4 with 1
IV: Replace the 1 on line 7 with 2

~~Which combination of o tions I t IV w al w th t o e rp ?~~

Just kidding, the teacher does not propose option I.

Which combination of options II to IV will allow the algorithm to do its intended purpose?

A: II and IV

B: III and IV

C: II and III

D: II and III and IV

Question 20

A minimum bottleneck path from a to b on an undirected weighted graph is a path between nodes a and b such that the minimum edge weight of any edge on the path is made as large as possible. A suitable algorithm for calculating minimum bottleneck path is:

A. Prim's

B. DFS

C. Bellman-Ford

D. Dijkstra's

ee. yay you reached the end of section A~

Have fun with Section B!!

SECTION B – fun section

Instructions for Section B

answer as many questions as you can! in the spaces provided! hopefully i added enough space haha.

master theorem? i have to explain it? ahaha… well hopefully i don't mess it up if i can just explain it right…

gc kg ax f > 1 lzwf qgmj l(f) kzgmdv tw dacw sl(f/t) + cf^u, jayzl, sfv lzwf kgew glzwj ugfklsfl v xgj f = 1 tml lzsl vgwkfl esllwj. kg dacw lqhausddq s, t, u, v, c sjw dacw zsnw lzwaj gof tgmfvk… tml a cafvs xgjygl lzwe kgjjq… al'k fgl dacw lzwq ywl lwklwv gf gj sfqlzafy… al'k bmkl ugeegf kwfkw, jwsddq. sfqosq kg dacw kgdmlagfk sjw mzz… dwl p = dgysjalze tskw t gx s jayzl, sfv lzwf ax p ak dwkk lzsf u, lzwf l(f) kgdmlagfk ak dacw g(f^u), sfv dacw… ax p ak wimsd lg u lzwf kgdmlagf ak dacw g(f^u dgy f), sfv zwjwk lzw jwsddq uggd hsjl, ax p ak yjwslwj lzsf u… lzwf kgdmlagf ak dacw g(f^p) zszs… vav lzsl cafvs escw kwfkw? szszs~ zghw a esvw s yggv wfgmyz wphdsfslagf~~ gz qwsz sfv kgjjq stgml lzw hzglg af lzsl dskl kwulagf… a osk hjwllq vwkhwjslw lg wkushw lzw ugfxafwk gx lzak wpse tml a kmhhgkw ak lzwjw sfq hgafl lg? a ewsf uggd hwghdw dacw qgm oadd vgofdgsv sfv vg lzak wpse… dacw a xwwd kg zmetdwv zszs

aaaaa i messed it up!!! sorry sorry sorry i'm sooo sorry~~ ;~;… i'm going to go back to hiding~~

Question 1 (7 marks)

The eccentric Arthur has invented a machine that can solve the halting problem. Given a program in the form of a Turing machine, it can determine in 1 second, with 100% accuracy, whether it will ever terminate.

a. The Goldbach Conjecture states that any even number greater than 2 can be expressed as the sum of two prime numbers. Explain how with this machine, you can determine the correctness of the Goldbach Conjecture in 1 second. (3 marks)

b. The halting problem is a problem within complexity class NP-Hard. Show that you can solve the Hamiltonian Path Problem for any given graph in 1 second using Arthur's machine. (2 marks)

____

____

____

____

____

c. Arthur tells you that his machine does not run a finite algorithm. Explain why this resolves the paradox present within the Halting problem. (2 marks)

____

____

____

____

____

Question 2 (6 marks)

Consider the following algorithm.

```
Algorithm FindN(N):
    // Input: A positive integer N
    // Note: floor(x) rounds the number x down to the nearest integer.
    If N ⩽ 1:
        return 0
    return FindN(floor(N/2) - 1) + FindN(floor(N/2)) + 10
```

a. Show that FindN has a worst case time complexity of O(N). (3 marks)

b. Describe how Dynamic Programming can be used to improve the time complexity for the FindN algorithm. (1 mark)

c. With Dynamic Programming, what is the new space complexity for the FindN algorithm? (1 mark)

d. With Dynamic Programming, what is the new time complexity for the FindN algorithm? (1 mark)

Question 3 (4 marks)

Ishraq has a list of N intervals of numbers over the number line. Each interval of numbers can be represented by a pair of integers (l, r), such that any number between l and r inclusive is in the interval. Write an O(N log N) algorithm to work out the total number of integers on the number line covered by one or more of these intervals.

i'm bored… haha...

Question 4 (9 marks)

Kevin has built a tourist park. The park consists of attractions and bidirectional roads. The speciality of the park is a puzzle, where people need to figure out the longest path through the attractions and roads such that no attraction is visited more than once. Kevin, being secretive, does not tell anybody the solution. At the park's opening day, tourists rush to work out the solution to his puzzle.

a. Explain how the Graph ADT can be used to model the park. (1 mark)

i mean, seriously, what kind of question is this?

b. A tourist wants to negate the edge weights and run the Bellman-Ford algorithm, reasoning that the shortest path when weights are inverted will be the longest path. Would this work? Justify your answer. (2 marks)

c. Is the problem NP-Hard? Explain why. (2 marks)

d. Some tourists also begin to consider approaches like Hill Climbing and simulated annealing. For both these approaches describe an approximate algorithm for this problem. (4 marks)

use as much space as you want!

i’ve saved the whole page for you~

Question 5 (11 marks)

Solve for T(N) in Big O notation for each of these recurrences.

Where required, use the substitution: N = 2^k.

a.

i. $T(N)=T(N-1)+O(N)$ (2 marks)

ii. $T(N)=N\times T(N-1)+O(1)$ (2 marks)

iii. $T(N)=\log_2(N)\times T(N/2)+O(1)$ (2 marks)

b.

i. $T(N)=2T(N/2)+O(N)$ (2 marks)

ii. $T(N)=T(\sqrt{N})+O(1)$ (1 mark)

iii. $T(N)=2T(\sqrt{N})+O(\log N)$ (2 marks)

huh? how are you meant to do this?

i should just shut up and let you do this exam haha...

Question 6 (13 marks)

Binary search is an algorithm that can be applied far outside its initial domain of searching sorted elements in a list.

Suppose Georgia has a *permutation* of size N (a list of all the integers from 1 to N (inclusive) in an arbitrary order).

a. How many possible such permutations are there? Express your answer in terms of N. (1 mark)

____

b. Suppose N = 9. Explain a possible way to assign a unique number to each permutation of size 9. (1 mark)

____

____

____

c.

i. Let H(N, A) be a function that takes in a number N and a permutation A, and returns a unique integer for each permutation of size N. Describe in pseudocode a possible algorithm for H. (3 marks)

ii. Suppose you had a list of permutations of size N. Explain how you could rearrange your list so you can use binary search to determine if a permutation was in your list. (1 mark)

d.

i. Let L be a list of all permutations of size N, sorted ascending by their H value. Show that, using binary search, you require $O(N \log N)$ iterations of binary search to determine the index of a permutation A in the list L. Where required, use Stirling's approximation for logarithm of factorials: $\ln(n!) = n \ln n - n + O(\ln n)$. (4 marks)

ii. Based in the intuition from part **i.**, explain why the lowest worst case time complexity of any algorithm for sorting a list of N integers using only comparisons (determining if a is greater than b) is $O(N \log N)$. (1 mark)

e.

i. The Fibonacci Heap is an implementation of priority queue with O(1) enqueue and O(log M) dequeue time complexities, where M is the number of elements in the priority queue. Using the Fibonacci heap, explain how you can sort a list of N elements in O(N log N). (1 mark)

ii. Based on the information given in part **d ii.** and **e i.**, explain why no priority queue can have O(1) enqueue and O(1) dequeue time complexities. (1 mark)

wow, only two lines?

better make your answer short~

more blank space~

Question 7 (14 marks)

You are given a list of coordinates of points on a 2D plane. You start at the first point in the list, and you have to reach all other points in the plane by moving between points. Define the distance between two points (x_1, y_1) and (x_2, y_2) as $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$.

a. Explain how you could model the relative distances between the points using a graph ADT. (2 marks)

b. Show how you can determine if 3 points lie on the same straight line without being given their coordinates, using their relative distances. (3 marks)

c. Prove by contradiction or otherwise that the sum of edge weights in a minimum spanning tree of the graph is less than or equal to the total length of the shortest Hamiltonian path of the graph. (3 marks)

d. With reference to Eulerian circuits, Explain why you can visit every edge in any tree graph exactly twice, and return to the node you started. (3 marks)

e. Hence, propose a polynomial time algorithm to find a Hamiltonian path of the graph that is at most twice as long as the shortest Hamiltonian path of the graph. (3 marks)

hey, i just read questions 8 to 12, and they're, like, super easy.

soooo~

i deleted them!

don't worry, you will still have a question to do~

i wrote a new question in their place, worth all of the remaining 16 marks :3

hope you like it! <3

question 8 (16 marks!)

soo like... i'm making cupcakes for you! i have already prepared exactly M unsprinkled cupcakes, and uhh i want to add an integer K grams of topping to each cupcake. for each cupcake, i only want to use at most 2 types of topping (cuz too many and the cupcake doesn't look as pretty :c), and uhh, i can add any integer amount of grams of topping to any cupcake.

for this, i have N types of topping, right, each with an integer number of grams of supply, and I've stored it all in a handy dandy list for you~

sooo i did the maths and there is just enough toppings in total to match the number of grams of sprinkles required for all the cupcakes. so like, the total number of grams of topping i have is equal to MK. can you help me, like, work out a way to give the cupcakes toppings? that would be great! ahaha~

a. suppose you’re given a solution to an instance of the problem. how would you model the solution using a graph ADT? (2 marks)

____

____

____

b. for this part, assume that M ≥ N-1

i. suggest a greedy approach to solving the problem if M ≥ N-1, that runs in O(MN) time complexity. you do not need to prove your approach yet! you do that in part **ii.** (2 marks)

____

____

____

____

____

____

____

ii. prove by induction that your approach works for M ≥ N-1 ahaha~ (2 marks)

iii. write pseudocode for your greedy algorithm! remember to output a graph as explained in part **a.**! (2 marks)

c. and now for this part, assume M = N-2

i. for M = N-2, explain how you can determine if there exists a subset S’ of topping types such that |S’|K – K = total supply of all ingredients in S’. (2 marks)

ii. the subset sum problem asks if in a set of integers, there exists a subset that adds to a certain value and the problem is, like, NP-Complete. Show that if M = N-2, my cupcake problem is also NP-Complete, by converting any instance of the subset sum problem to an instance of the cupcake problem. (3 marks)

d. assume M ≥ N-2. describe an algorithm that solves this problem in $O(MN + NK^2)$ worst case time complexity! you may refer to your answers from parts **b.** and **c.** (3 marks)

END OF QUESTION AND ANSWER BOOK

you did it! you made it through the exam! ahaha~

uhh…

hope you enjoyed it~

goodbyeeee~~

…vocaloid?

oh yeah i also sing from time to time

ahaha~

Special Thanks:
Quang Ong for proofreading
Chinese National Olympiad in Informatics for providing inspiration for Q8
1st Place (and IA) for sponsoring International Olympiad in Informatics 2018