Algorithms and Satisfiability

(DAT6)

Exam Assignments

10.00 - 13.00, 1 June 2021

Full name:	
Student number:	
E-mail at student.aau.dk:	

This exam consists of two parts of exercises. Exercise 1 is for Algorithms (the A-Part). Exercise 2 is for Satisfiability (the S-Part).

During the exam you are allowed to consult the textbook, notes, and slides (digital versions are also allowed). However, you are NOT allowed to use your computer to search for any other materials on the internet.

Submission to digital exam: upload scans or photos of your answers made on the exam papers or/and other separate papers.

Exercise 1, The A-Part, 50 points in total

1. Given a directed, weighted graph G, we run Floyd-Warshall algorithm on G. After the first iteration ($k = 1$) of the Floyd-Warshall algorithm, the **distance matrix** $D^{(1)}$ and **predecessor matrix** $P^{(1)}$ look like the following.

$$D^{(1)} = \begin{pmatrix} 0 & \infty & \infty & \infty \\ 1 & 0 & \infty & 2 \\ \infty & 2 & 0 & \infty \\ -4 & \infty & \infty & 0 \end{pmatrix} P^{(1)} = \begin{pmatrix} Nil & Nil & Nil & Nil \\ 2 & Nil & Nil & 2 \\ Nil & 3 & Nil & Nil \\ 4 & Nil & Nil & Nil \end{pmatrix}$$

1.1 (*4 points*) Could the following matrix A be the adjacency matrix of graph G?

$$A = \begin{pmatrix} 0 & \infty & \infty & \infty \\ 1 & 0 & \infty & 2 \\ \infty & 2 & 0 & \infty \\ -4 & \infty & \infty & 0 \end{pmatrix}$$

☐ **a)** Yes, it is possible. ☐ **b)** No, it is impossible.

1.2 (*8 points*) Write down the distance matrix $D^{(2)}$ and predecessor matrix $P^{(2)}$, i.e., after the second iteration ($k = 2$) of the Flord-Warshall algorithm.

$$D^{(2)} = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix} \quad P^{(2)} = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

2. We want to sort many numbers that are stored in a file that takes 20,000 disk pages. We use a computer with 5 pages of main memory to perform external memory merge sort. Note that the disk page and the main memory page are of the same size.

2.1 (*3 points*) How many runs will the external memory merge sort produce on the first phase?

☐ **a)** 5 ☐ **b)** 4,000 ☐ **c)** 20,000
☐ **d)** $\lceil \log_4(4000) \rceil$=6 ☐ **e)** $\lceil \log_5(20000) \rceil$=7 ☐ **f)** $\lceil \log_4(20000) \rceil$=8

2.2 (*3 points*) Can this computer afford a 5-way external memory merge sort?

☐ **a)** Yes. ☐ **b)** No.

2.3 (*4 points*) In the second phase, how many iterations does the 4-way external memory merge-sort need to run?

Write down the number or equation here _____.

2.4 (*4 points*) We consider either a disk page read or a disk page write as an I/O operation. How many I/O operations are performed by the 4-way external memory merge-sort?

Write down the number or equation here _____.

3.1 (*4 points*) Consider an approximation algorithm with approximation ratio 1.5 for solving a NP-complete problem Q. Assume that Q is a **maximization** problem and its optimal solution is 120. What is the **worst** value that the approximation algorithm can return?

☐ **a)** 100 ☐ **b)** 120

☐ **c)** 80 ☐ **d)** 75

3.2 (*4 points*) Consider an approximation algorithm with approximation ratio 2.5 for solving a NP-complete problem Q. Assume that Q is a **minimization** problem and its optimal solution is 60. What is the **worst** value that the approximation algorithm can return?

☐ **a)** 50 ☐ **b)** 100

☐ **c)** 120 ☐ **d)** 150

4. We initialize a stack of size k and in the beginning the stack is empty. We start performing a sequence of *PUSH* and *POP* operations. After every k *PUSH* and *POP* operations, we make a copy of the entire stack so far to make a backup using a *COPY* operation.

4.1 (*6 points*) What is the worst case asymptotic cost of a *PUSH*, a *POP*, and a *COPY* operation, respectively.

4.2 (*10 points*) What is the asymptotic cost of n stack operations (including *PUSH*, *POP*, and *COPY*) and what is the amortized cost per each operations? Please elaborate how do you come to the results.

Exercise 2, The S-Part, 50 points in total

1. Consider the set of clauses, $\Delta = \{\{S, \neg R\}, \{\neg S, Q, \neg P\}, \{\neg Q\}, \{S, Q, \neg P\}$.

1.1 (6 points) Run DPLL and give the trace until you get to the first conflict. In the splitting rule, always select the unassigned proposition that comes first in the alphabet, and try first truth value T then truth value F.

Rule applied _____

Result _____

Rule applied _____

Result _____

Rule applied _____

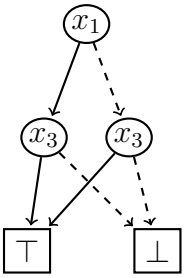
Result _____

1.2 (6 points) Draw the implication graph for the conflict and all conflict graphs induced by the implication graph.

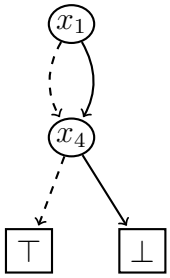
1.3 (2 points) Which is the clause that we can learn with the clause learning method presented in the lecture?

Learned Clause _____

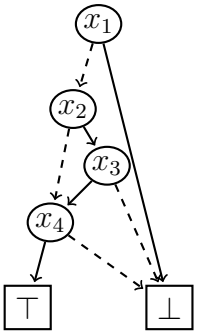
2.1 (12 points) Some of the following ROBDDs are not correct. For each of them, say if it is correct or not and justify your answer:



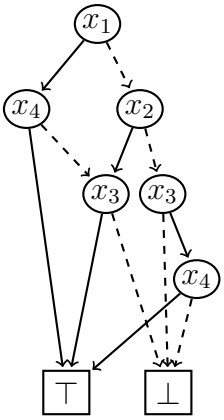
Answer



Answer



Answer



Answer

2.2 (3 points) Supose that we have three BDDs A, B, and C having 11 nodes each. If we use the apply operation twice in order to compute the BDD $(A \wedge B) \wedge C$, the resulting BDD **could** have (mark all correct options):

☐ 1 node ☐ 10 nodes ☐ 100 nodes ☐ 1000 nodes ☐ 10000 nodes

3 Planning

3.1 (6 points) Consider the following pairs of heuristics and tick the boxes where the dominance relation holds; where a heuristic h dominates h' if $h(s) \geq h'(s)$ for all states s in all planning tasks. Heuristics are incomparable if there exist states s and s' so that $h(s) > h'(s)$ and $h'(s') > h(s')$. Mark the correct option.

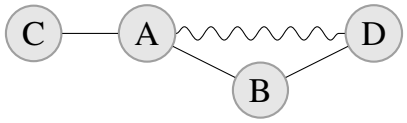
A) ☐ h^+ dominates h^* ☐ h^* dominates h^+ ☐ h^* and h^+ are incomparable

B) ☐ h^{FF} dominates h^* ☐ h^* dominates h^{FF} ☐ h^* and h^{FF} are incomparable

C) ☐ h^{FF} dominates h^+ ☐ h^+ dominates h^{FF} ☐ h^{FF} and h^+ are incomparable

3.2 (5 points) Suppose that you want to prove that the optimal solution for a planning task Π has 10 actions using Planning as SAT. Which formulas would you construct? For each formula specify (a) the number of steps, (b) whether sequential or $\forall$-step semantics are used, and (c) what is the expected result for the SAT solver (satisfiable or unsatisfiable), provided that the optimal solution has indeed 10 actions.

4 (10 points) Consider the following map with four different locations: A, B, C, D. There are two different types of connections: footways (denoted by straight lines), and cycleways (denoted by curved lines). It is not allowed to use the cycleways without wearing a bicycle helmet. Alice wants to get from location A to location D, but she is not wearing the helmet.



Using the following predicates, fill the PDDL snippets that specify the ride-bike action (there are more actions, but we ignore them here), the objects, initial state and goal:

```
(:predicates
      (at ?l) % Describes the position of Alice
      (wearing-helmet)
      (footway ?l1 ?l2)
      (cycleway ?l1 ?l2)
)

(:action ride-bike

)

(:objects

)

(:init


)

(:goal


)
```